FORGEROCK

API Guide

/ Autonomous Identity 2020.6.4

Latest update: 2020.6.4

,
, ,

Abstract

This guide is targeted to developers who need to access Autonomous Identity using the REST Application Programming Interface (API).

Table of Contents

Overview iv
1. Introduction to the Autonomous Identity API 1
    Swagger 1
    Postman Collections 2
2. Configurations 3
3. Authentication 29
4. User Details 31
5. Config 35
6. Report 36
7. Company View 43
8. Access Control 48
9. Single View with Application 53
10. Role Owner with Application Oriented 56
11. Manager with Application Oriented 59
12. Entitlements 62
Glossary 65

Overview

This guide is targeted to developers who want to access Autonomous Identity using the REST Application Programming Interface (API).

ForgeRock® Autonomous Identity is an entitlements analytics system that lets you fully manage your company's access to your data.

An entitlement refers to the rights or privileges assigned to a user or thing for access to specific resources. A company can have millions of entitlements without a clear picture of what they are, what they do, and who they are assigned to. Autonomous Identity solves this problem by using advanced artificial intelligence (AI) and automation technology to determine the full entitlements landscape for your company. The system also detects potential risks arising from incorrect or over-provisioned entitlements that lead to policy violations. Autonomous Identity eliminates the manual re-certification of entitlements and provides a centralized, transparent, and contextual view of all access points within your company.

Quick Start

Authentication Learn how to access the Authentication endpoints.	Config Learn about the config endpoint.	Configuration Learn about the configuration endpoint.
User Details Learn about the user details endpoint.	Reports Learn about the Reports API.	Access Control Learn about the access control endpoints.
Single View with App Learn about the single view with app endpoints.	Role Owner with App Learn how to set the role owner with app endpoints.	Manager with App Learn how to set the manager with app endpoints.

Chapter 1
Introduction to the Autonomous Identity API

Autonomous Identity provides a RESTful application programming interface (API) that lets you use HTTP request methods (GET, PUT, and POST) to interact with the system and its components. The API lets a developer make requests to send or receive data to an Autonomous Identity endpoint, a point where the API communicates with the system. The data that is sent or returned is in JavaScript Object Notation (JSON) format.

Autonomous Identity provides easy integration with popular API clients, such as Swagger and Postman. Autonomous Identity also provides easy integration with Swagger by providing a `swagger.yml` file with the deployment. You can download and install the Autonomous Identity Postman Collection from ForgeRock Google Cloud Repository (gcr.io).

The next two sections show you how to set up Swagger and Postman.

- "Swagger"
- "Postman Collections"

Swagger

The Autonomous Identity API supports Swagger, a popular API design tool that allows you to interact with the API and the Configuration Service API. You can access the API using the Swagger client at `https://<autoid-ui>/swagger`. For example, `https://autoid-ui.forgerock.com/swagger`.

The API microservice serves a `swagger.yml` file, accessible at `https://<autoid-ui>/api/swagger`. For example, `https://autoid-ui.forgerock.com/api/swagger`.

The Configuration services also serves the `swagger.yml` file, accessible at `https://<autoid-ui>/conf/swagger`. For example, `https://autoid-ui.forgerock.com/conf/swagger`.

Access the Autonomous Identity API on Swagger

1. Open a browser and point it to `https://<autoid-ui>/swagger`.
2. To view the Autonomous Identity API, enter `<autoid-ui>/api/swagger`. For example, enter `<autoid-ui>.forgerock.com/api/swagger`, and then click Explore.
3. First, obtain a bearer token by accessing the `/api/authentication/login` endpoint. You provide your username and password on Autonomous Identity in the request body.

4. Upon a successful authentication, the response returns a bearer token.

5. Use the bearer token in the Authorize section of the Swagger client.

6. Once you are authorized, access the API endpoints to see or add data. For example, access the Autonomous Identity Company View endpoint to get information on the company dataset.

Access the Autonomous Identity Configuration Service API on Swagger

1. On the Swagger client, click Authorize and then log in using the Configuration Service admin username and password.

2. Once you are authorized, access the API endpoints to see or add data. For example, access the Autonomous Identity configuration endpoint to get information.

Postman Collections

ForgeRock provides a Postman Collection that lets you use saved requests to access the Autonomous Identity API endpoints.

You need to authenticate first by calling the Authentication/Login request before doing any other requests as the token is required to access the endpoints. By default, there is a post request runner that will pull the token out and set it into the environment variable {{token}}, and it will then be passed into the subsequent API calls to the other sections of the API as a Bearer token.

When creating new requests, if they are created in an existing folder, they will inherit the Bearer Auth settings from the folder itself. New folders will need to have these authorization settings set into them with Bearer auth and {{token}} as the value.

The format of the API variable should follow `${domain}/api/`. It should always have the suffix of `api/` no matter the domain.

Set Up the Autonomous Identity Postman Collection

1. Download the Autonomous Identity Postman Collections from the BackStage.

2. Open Postman.

3. Click Import, click the box, and select `API.postman_collection.json`.

4. To set the environment, click Import, click the box, and select `postman_environment.json`.

Chapter 2
Configurations

The following are Autonomous Identity configuration endpoints:

PUT RevokeCertifyAccessConf

Sets the schema definition for the matching database table (`revoke_certify_access_request`), which is stored in Consul. This endpoint allows the configuration to be changed on the fly.

Endpoint

```
{{conf_url}}/RevokeCertifyAccessConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "RevokeCertifyAccess",
  "modelDefinition": {
    "fields": {
      "is_processed": "boolean",
      "entitlement": "text",
      "user": "text",
      "manager": "text",
      "manager_decision": "int",
      "manager_date_created": "timestamp",
      "role_owner": "text",
      "role_owner_decision": "int",
      "role_owner_date_created": "timestamp",
      "date_created": "timestamp"
   },
   "key": [
      "is_processed"
   ],
   "table_name": "revoke_certify_access_request"
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/RevokeCertifyAccessConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name" : "RevokeCertifyAccess",
  "modelDefinition": {
    "fields": {
      "is_processed": "boolean",
      "entitlement": "text",
      "user": "text",
      "manager": "text",
      "manager_decision": "int",
      "manager_date_created": "timestamp",
      "role_owner": "text",
      "role_owner_decision": "int",
      "role_owner_date_created": "timestamp",
      "date_created": "timestamp"
   },
   "key": [
      "is_processed"
   ],
   "table_name": "revoke_certify_access_request"
  }
}'
```

PUT CompanyViewOverviewConf

Sets the schema definition for the related database table {{company_view_overview}}.

Endpoint

```
{{conf_url}}/CompanyViewOverviewConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "CompanyViewOverview",
  "modelDefinition": {
    "fields": {
      "key": "text",
      "total_employees": "int",
      "employees_wo_manager": "int",
      "employees_w_manager": "int",
      "entitlements_without_roleowners": "int",
      "entitlements_with_roleowners": "int",
      "total_entitlements": "int",
      "entitlements_covered_by_model": "int",
      "entitlements_not_covered": "int",
      "entitlements_w_no_users": "int",
      "entitlements_w_one_user": "int",
      "entitlements_w_zero_to_five_users": "int",
      "entitlements_w_five_to_ten_users": "int",
      "entitlements_w_ten_to_hundred_users": "int",
      "entitlements_w_hundred_to_onek_user": "int",
      "entitlements_w_onek_to_tenk_users": "int",
      "entitlements_w_tenk_users": "int",
      "entitlements_w_hundredk_users": "int"
    },
    "key": [
      "key"
    ],
    "table_name": "company_view_overview"
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/CompanyViewOverviewConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "CompanyViewOverview",
  "modelDefinition": {
    "fields": {
      "key": "text",
      "total_employees": "int",
      "employees_wo_manager": "int",
      "employees_w_manager": "int",
      "entitlements_without_roleowners": "int",
      "entitlements_with_roleowners": "int",
      "total_entitlements": "int",
      "entitlements_covered_by_model": "int",
      "entitlements_not_covered": "int",
      "entitlements_w_no_users": "int",
      "entitlements_w_one_user": "int",
      "entitlements_w_zero_to_five_users": "int",
      "entitlements_w_five_to_ten_users": "int",
      "entitlements_w_ten_to_hundred_users": "int",
      "entitlements_w_hundred_to_onek_user": "int",
      "entitlements_w_onek_to_tenk_users": "int",
      "entitlements_w_tenk_users": "int",
      "entitlements_w_hundredk_users": "int"
    },
    "key": [
      "key"
    ],
    "table_name": "company_view_overview"
  }
}'
```

PUT CompanyViewEmployeeTypeConf

>Sets the schema definition for the related database table {{company_view_employeetype}}.

Endpoint

```
{{conf_url}}/CompanyViewEmployeeTypeConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "CompanyViewEmployeeType",
  "modelDefinition": {
    "fields": {
      "type": "text",
      "high": "int",
      "medium": "int",
      "low": "int",
      "null_conf": "int",
      "total": "int"
    },
    "key": [
      "type"
    ],
    "table_name": "company_view_employee_type"
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/CompanyViewEmployeeTypeConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "CompanyViewEmployeeType",
  "modelDefinition": {
    "fields": {
      "type": "text",
      "high": "int",
      "medium": "int",
      "low": "int",
      "null_conf": "int",
      "total": "int"
    },
    "key": [
      "type"
    ],
    "table_name": "company_view_employee_type"
  }
}'
```

PUT EntitlementAverageConfScoreConf

Sets the schema definition for the related database table {{entitlement_average_conf_score}}.

Endpoint

```
{{conf_url}}/EntitlementAverageConfScoreConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "EntitlementAverageConfScore",
 "modelDefinition": {
    "fields": {
       "org": "text",
       "avg_score": "float",
       "entitlement": "text"
    },
    "key": [
       "org"
    ],
    "table_name": "entitlement_average_conf_score"
 }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/EntitlementAverageConfScoreConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "EntitlementAverageConfScore",
 "modelDefinition": {
    "fields": {
       "org": "text",
       "avg_score": "float",
       "entitlement": "text"
    },
    "key": [
       "org"
    ],
    "table_name": "entitlement_average_conf_score"
 }
}'
```

PUT EntitlementUserScoresConf

Sets the schema definition for the related database table {{entitlement_user_scores}}.

Endpoint

```
{{conf_url}}/EntitlementUserScoresConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "EntitlementUserScores",
    "modelDefinition": {
      "fields": {
        "entitlement": "text",
        "entitlement_name": "text",
        "user": "text",
        "user_name": "text",
        "score": "float",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text"
      },
      "key": ["entitlement"],
      "table_name": "entitlement_user_scores"
    }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/EntitlementUserScoresConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "EntitlementUserScores",
    "modelDefinition": {
      "fields": {
        "entitlement": "text",
        "entitlement_name": "text",
        "user": "text",
        "user_name": "text",
        "score": "float",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text"
      },
      "key": ["entitlement"],
      "table_name": "entitlement_user_scores"
    }
  }'
```

PUT GraphByRoleConf

Sets the schema definition for the related database table {{graph_by_role}}.

Endpoint

```
{{conf_url}}/GraphByRoleConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "GraphByRole",
    "modelDefinition": {
      "fields": {
        "role": "text",
        "entitlement": "text",
        "entitlement_name": "text",
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["role"],
      "table_name": "graph_by_role"
    }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/GraphByRoleConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "GraphByRole",
    "modelDefinition": {
      "fields": {
        "role": "text",
        "entitlement": "text",
        "entitlement_name": "text",
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["role"],
      "table_name": "graph_by_role"
    }
  }'
```

PUT GraphConf

Sets the schema definition for the related database table {{graph}}.

Endpoint

```
{{conf_url}}/GraphConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "Graph",
    "modelDefinition": {
      "fields": {
        "manager": "text",
        "user": "text",
        "manager_name": "text",
        "user_name": "text"
      },
      "key": ["manager"],
      "table_name": "graph_by_manager"
    }
  }
```

Example Request

```
curl --location --request PUT '{{conf_url}}/GraphConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "Graph",
    "modelDefinition": {
      "fields": {
        "manager": "text",
        "user": "text",
        "manager_name": "text",
        "user_name": "text"
      },
      "key": ["manager"],
      "table_name": "graph_by_manager"
    }
  }'
```

PUT ManagerConf

Set manager data.

Endpoint

```
{{conf_url}}/ManagerConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "Manager",
  "modelDefinition": {
    "fields": {
      "org": "text",
      "manager": "int"
    },
    "key": [
      "manager"
   ],
    "table_name": "managers_by_org"
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/ManagerConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "Manager",
  "modelDefinition": {
    "fields": {
      "org": "text",
      "manager": "int"
    },
    "key": [
      "manager"
    ],
    "table_name": "managers_by_org"
  }
}'
```

PUT RoleOwnerConf

Sets the schema definition for the related database table {{role_owner}}.

Endpoint

```
{{conf_url}}/RoleOwnerConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "RoleOwner",
    "modelDefinition": {
      "fields": {
        "role": "text",
        "role_name": "text",
        "entitlement": "text",
        "entitlement_name": "text",
        "user": "text",
        "user_name": "text",
        "score": "float",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["role"],
      "table_name": "usr_scores_by_role"
    }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/RoleOwnerConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "RoleOwner",
    "modelDefinition": {
      "fields": {
        "role": "text",
        "role_name": "text",
        "entitlement": "text",
        "entitlement_name": "text",
        "user": "text",
        "user_name": "text",
        "score": "float",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["role"],
      "table_name": "usr_scores_by_role"
    }
  }'
```

PUT RoleOwnerWithAppConf

Set role owner with application.

Endpoint

```
{{conf_url}}/RoleOwnerWithAppConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "RoleOwnerWithApp",
  "modelDefinition": {
    "fields": {
      "role": "text",
      "role_name": "text",
      "entitlement": "text",
      "entitlement_name": "text",
      "user": "text",
      "user_name": "text",
      "score": "float",
      "justification": {
         "type": "list",
         "typeDef": "<text>"
      },
      "app_id": "text",
      "app_name": "text"
    },
    "key": [
      "role"
    ],
    "table_name": "usr_scores_by_role_with_app"
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/RoleOwnerWithAppConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "RoleOwnerWithApp",
  "modelDefinition": {
    "fields": {
      "role": "text",
      "role_name": "text",
      "entitlement": "text",
      "entitlement_name": "text",
      "user": "text",
      "user_name": "text",
      "score": "float",
      "justification": {
         "type": "list",
         "typeDef": "<text>"
      },
      "app_id": "text",
      "app_name": "text"
    },
    "key": [
      "role"
    ],
    "table_name": "usr_scores_by_role_with_app"
  }
}'
```

PUT UserConf

Sets the schema definition for the related database table {{user}}.

Endpoint

```
{{conf_url}}/UserConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "User",
    "modelDefinition": {
      "fields": {
        "user": "text",
        "manager": "text",
        "department": "text",
        "empType": "text",
        "udfChief": "text",
        "udfCostCenter": "text",
        "jobCode": "text",
        "buildingCode": "text",
        "lob": "text",
        "lobSubgroup": "text",
        "userName": "text",
        "managerName": "text",
        "departmentDescription": "text",
        "jobDescription": "text"
      },
      "key": ["user"],
      "table_name": "user"
    }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/UserConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "User",
    "modelDefinition": {
      "fields": {
        "user": "text",
        "manager": "text",
        "department": "text",
        "empType": "text",
        "udfChief": "text",
        "udfCostCenter": "text",
        "jobCode": "text",
        "buildingCode": "text",
        "lob": "text",
        "lobSubgroup": "text",
        "userName": "text",
        "managerName": "text",
        "departmentDescription": "text",
        "jobDescription": "text"
      },
      "key": ["user"],
      "table_name": "user"
    }
}'
```

PUT UserScoreConf

Sets the schema definition for the related database table {{user_score}}.

Endpoint

```
{{conf_url}}/UserScoreConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "name": "UserScore",
    "modelDefinition": {
      "fields": {
        "manager": "text",
        "user": "text",
        "manager_name": "text",
        "user_name": "text",
        "score": "float",
        "entitlement": "text",
        "entitlement_name": "text",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["manager"],
      "table_name": "usr_scores_by_manager"
    }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/UserScoreConf' \
--header 'Content-Type: application/json' \
--data-raw '{
    "name": "UserScore",
    "modelDefinition": {
      "fields": {
        "manager": "text",
        "user": "text",
        "manager_name": "text",
        "user_name": "text",
        "score": "float",
        "entitlement": "text",
        "entitlement_name": "text",
        "justification": {
          "type": "list",
          "typeDef": "<text>"
        },
        "app_id": "text",
        "app_name": "text",
        "high_risk": "text"
      },
      "key": ["manager"],
      "table_name": "usr_scores_by_manager"
    }
  }'
```

PUT FilteringOptionsModelConf

Sets the schema definition for the related database table {{filtering_options_model}}.

Endpoint

```
{{conf_url}}/FilteringOptionsModelConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "FilteringOptions",
 "modelDefinition": {
     "fields":{
         "type": "int",
         "owner_id"    : "text",
         "group" : "text",
         "id" : "text",
         "name"     : "text",
         "user_ids": {
           "type": "list",
           "typeDef": "<text>"
        }
       },
       "key": ["type"],
       "table_name": "filtering_options"
 }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/FilteringOptionsModelConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "FilteringOptions",
 "modelDefinition": {
     "fields":{
         "type": "int",
         "owner_id"    : "text",
         "group" : "text",
         "id" : "text",
         "name"     : "text",
         "user_ids": {
           "type": "list",
           "typeDef": "<text>"
        }
       },
       "key": ["type"],
       "table_name": "filtering_options"
 }
}'
```

PUT CompanyViewMostCriticalEnttConf

Sets the schema definition for the related database table {{company_view_most_critical_entt}}.

Endpoint

```
{{conf_url}}/CompanyViewMostCriticalEnttConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "name": "CompanyViewMostCriticalEntt",
 "modelDefinition": {
   "fields":{
       "org": "text",
        "entt_id"    : "text",
        "entt_name" : "text",
       "high" : "int",
       "medium" : "int",
       "seq" : "int",
        "low": "int",
        "total_employees" : "int",
       "avg_conf_score": "float"
   },
   "key": ["org"],
   "table_name": "company_view_most_critical_entt"
 }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/CompanyViewMostCriticalEnttConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "name": "CompanyViewMostCriticalEntt",
 "modelDefinition": {
    "fields":{
       "org": "text",
       "entt_id"    : "text",
       "entt_name" : "text",
       "high" : "int",
        "medium" : "int",
        "seq" : "int",
       "low": "int",
       "total_employees" : "int",
       "avg_conf_score": "float"
     },
    "key": ["org"],
    "table_name": "company_view_most_critical_entt"
 }
}'
```

PUT FilteringOptionsConf

Set the filtering options.

Endpoint

```
{{conf_url}}/FilteringOptionsConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
      "filteringOptions": [
        {
            "type": "user",
            "groupName": "STATE",
            "title": "State"
        },
        {
            "type": "user",
            "groupName": "DEPARTMENT",
            "title": "Department"
        }
      ]
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/FilteringOptionsConf' \
--header 'Content-Type: application/json' \
--data-raw '{
      "filteringOptions": [
        {
            "type": "user",
            "groupName": "STATE",
            "title": "State"
        },
        {
            "type": "user",
            "groupName": "DEPARTMENT",
            "title": "Department"
        }
      ]
  }'
```

PUT OrgNameConf

Set the organization name.

Endpoint

```
{{conf_url}}/OrgNameConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "orgName": "abc"
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/OrgNameConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "orgName": "abc"
}'
```

PUT ConfidenceScoreThreholdsConf

Set the confidence score thresholds.

Endpoint

```
{{conf_url}}/ConfidenceScoreThreholdsConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "thresholds": {
   "top": 1.01,
   "high": 0.75,
   "medium": 0.35,
   "low": 0
 }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/ConfidenceScoreThresholdsConf' \
--header 'Content-Type: application/json' \
--data-raw '{
 "thresholds": {
   "top": 1.01,
   "high": 0.75,
   "medium": 0.35,
   "low": 0
 }
}'
```

PUT UIHRData

Set the UI HR data.

Endpoint

```
{{conf_url}}/UIHRData
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
  "user": "User",
  "manager": "Manager",
  "emptype": " Employee Type",
  "buildingcode": "Building Code",
  "department": "Department Code",
  "departmentdescription": "Department Description",
  "jobcode": "Job code",
  "jobdescription": "Job Code Description",
  "lob": "Line Of Business",
  "lobsubgroup": "Line Of Business SubGroup",
  "managername": "Manager Name",
  "udfchief": "UDF Chief",
  "udfcostcenter": "UDF Cost Center",
  "username": "User Name"
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/UIHRData' \
--header 'Content-Type: application/json' \
--data-raw '{
  "user": "User",
  "manager": "Manager",
  "emptype": " Employee Type",
  "buildingcode": "Building Code",
  "department": "Department Code",
  "departmentdescription": "Department Description",
  "jobcode": "Job code",
  "jobdescription": "Job Code Description",
  "lob": "Line Of Business",
  "lobsubgroup": "Line Of Business SubGroup",
  "managername": "Manager Name",
  "udfchief": "UDF Chief",
  "udfcostcenter": "UDF Cost Center",
  "username": "User Name"
}'
```

PUT UIJustifications

Set the UI justifications.

Endpoint

```
{{conf_url}}/UIJustifications
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
  "USR_MANAGER_KEY": "Supervisor",
  "USR_DEPT_NO": "Department No",
  "USR_EMP_TYPE": "Employee Type",
  "USR_UDF_CHIEF": " UDF Chief",
  "USR_UDF_COST_CENTER": "UDF Cost Center",
  "USR_UDF_JOBCODE": "Job Code",
  "USR_UDF_BUILDINGCODE": "Building Code",
  "USR_UDF_LOB": "Line Of Business",
  "USR_UDF_LOBSUBGROUP": "Line of Business Subgroup"
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/UIJustifications' \
--header 'Content-Type: application/json' \
--data-raw '{
  "USR_MANAGER_KEY": "Supervisor",
  "USR_DEPT_NO": "Department No",
  "USR_EMP_TYPE": "Employee Type",
  "USR_UDF_CHIEF": " UDF Chief",
  "USR_UDF_COST_CENTER": "UDF Cost Center",
  "USR_UDF_JOBCODE": "Job Code",
  "USR_UDF_BUILDINGCODE": "Building Code",
  "USR_UDF_LOB": "Line Of Business",
  "USR_UDF_LOBSUBGROUP": "Line of Business Subgroup"
}'
```

PUT HighRiskConf

Set the high risk filter value.

Endpoint

```
{{conf_url}}/HighRiskConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
  "filterValue": "1"
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/HighRiskConf' \
--header 'Content-Type: application/json' \
--data-raw '{
  "filterValue": "1"
}'
```

PUT JustificationDelimiter

Set the justification delimiter to separate the different justifications in the string saved in Cassandra. For .csv files, the delimiter is a comma (,).

Endpoint

```
{{conf_url}}/JustificationDelimiter
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
    "justificationDelimeter": "_"
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/JustificationDelimeter' \
--header 'Content-Type: application/json' \
--data-raw '{
    "justificationDelimeter": "_"
  }'
```

PUT PermissionsConf

Set the permissions.

Endpoint

```
{{conf_url}}/PermissionsConf
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

Content-Type	**application/json**

Body

```
{
  "actions": [
    "CERTIFY__ENTITLEMENTS_TO_USERS",
    "CERTIFY__USERS_TO_ENTITLEMENTS",
    "FILTER__ENTITLEMENTS",
    "REVOKE__CERTIFY_ACCESS",
    "SEARCH__USER",
    "SEARCH__USER_ENTITLEMENTS",
    "SEARCH__SUPERVISOR_USER_ENTITLEMENTS",
    "SHOW__ASSIGNMENTS_STATS",
    "SHOW__COMPANY_PAGE",
    "SHOW__COMPANY_COVERAGE_DATA",
    "SHOW__COMPANY_ENTITLEMENTS_DATA",
    "SHOW__COMPANY_EMPLOYEE_PAGE",
    "SHOW__CRITICAL_ENTITLEMENTS",
    "SHOW__EMPLOYEE",
    "SHOW__ENTITLEMENT",
    "SHOW__ENTITLEMENT_AVG_GROUPS",
    "SHOW__ENTITLEMENT_AVG_GROUP_DETAILS",
    "SHOW__ENTITLEMENT_USERS",
    "SHOW__FILTER_OPTIONS",
    "SHOW__ROLE_OWNER_PAGE",
    "SHOW__ROLE_OWNER_USER_DATA",
    "SHOW__ROLE_OWNER_ENT_DATA",
    "SHOW__ROLE_OWNER_AUTO_DATA",
    "SHOW__SUPERVISOR_PAGE",
    "SHOW__SUPERVISOR_DETAILS_PAGE",
    "SHOW__SUPERVISOR_ENT_DATA",
    "SHOW__SUPERVISOR_USER_DATA",
    "SHOW__SUPERVISOR_ENTITLEMENT_USERS",
    "SHOW__SUPERVISOR_USER_ENTITLEMENTS",
    "SHOW__ROLEOWNER_UNSCORED_ENTITLEMENTS",
    "SHOW__SUPERVISOR_UNSCORED_ENTITLEMENTS",
    "SHOW__UNSCORED_ENTITLEMENTS",
    "SHOW__USER",
    "SHOW__ALL_ROLE_OWNER_DATA"
  ],
  "permissions": {
    "Zoran Admin": {
      "can": "*"
    },
    "Zoran Entitlement Owner": {
      "can": [
        "FILTER__ENTITLEMENTS",
        "SEARCH__USER_ENTITLEMENTS",
        "SHOW__ENTITLEMENT",
        "SHOW__ENTITLEMENT_USERS",
        "SHOW__FILTER_OPTIONS",
        "SHOW__ROLEOWNER_UNSCORED_ENTITLEMENTS",
        "SHOW__ROLE_OWNER_PAGE",
        "SHOW__ROLE_OWNER_ENT_DATA",
        "SHOW__ROLE_OWNER_AUTO_DATA",
        "SHOW__ROLE_OWNER_USER_PAGE",
        "SHOW__ROLE_OWNER_ENT_PAGE",
        "SHOW__USER_ENTITLEMENTS",
        "SHOW__UNSCORED_ENTITLEMENTS",
        "CERTIFY__ENTITLEMENTS_TO_USERS",
        "CERTIFY__USERS_TO_ENTITLEMENTS",
        "REVOKE__CERTIFY_ACCESS"
      ]
    },
    "Zoran Executive": {
      "can": [
        "SHOW__ASSIGNMENTS_STATS",
        "SHOW__COMPANY_PAGE",
        "SHOW__COMPANY_COVERAGE_PAGE",
        "SHOW__COMPANY_ENTITLEMENTS_PAGE",
        "SHOW__COMPANY_EMPLOYEE_PAGE",
        "SHOW__CRITICAL_ENTITLEMENTS",
        "SHOW__ENTITLEMENT_AVG_GROUPS",
        "SHOW__ENTITLEMENT_AVG_GROUP_DETAILS",
        "SHOW__USER_ENTITLEMENTS"
      ]
    },
    "Zoran Supervisor": {
      "can": [
        "FILTER__ENTITLEMENTS",
        "SHOW__EMPLOYEE",
        "SHOW__FILTER_OPTIONS",
        "SHOW__SUPERVISOR_PAGE",
        "SHOW__SUPERVISOR_DETAILS_PAGE",
        "SHOW__SUPERVISOR_ENT_DATA",
        "SHOW__SUPERVISOR_USER_DATA",
        "SHOW__SUPERVISOR_ENTITLEMENT_USERS",
        "SHOW__SUPERVISOR_USER_ENTITLEMENTS",
        "SEARCH__SUPERVISOR_USER_ENTITLEMENTS",
        "SHOW__SUPERVISOR_UNSCORED_ENTITLEMENTS",
        "CERTIFY__ENTITLEMENTS_TO_USERS",
        "CERTIFY__USERS_TO_ENTITLEMENTS",
        "REVOKE__CERTIFY_ACCESS"
      ]
    },
    "Zoran User": {
      "can": [
        "SHOW__CERTIFICATIONS",
        "SEARCH__USER",
        "SHOW__ENTITLEMENT",
        "SHOW__USER"
      ]
    }
  }
}
```

Example Request

```
curl --location --request PUT '{{conf_url}}/PermissionsConf' \
--header 'Content-Type: application/json' \
--data-raw '{"actions":
["CERTIFY__ENTITLEMENTS_TO_USERS","CERTIFY__USERS_TO_ENTITLEMENTS","FILTER__ENTITLEMENTS","REVOKE__CERTIFY_ACCE
{"Zoran Admin":{"can":"*"},"Zoran Entitlement Owner":{"can":
["FILTER__ENTITLEMENTS","SEARCH__USER_ENTITLEMENTS","SHOW__ENTITLEMENT","SHOW__ENTITLEMENT_USERS","SHOW__FILTER
 Executive":{"can":
["SHOW__ASSIGNMENTS_STATS","SHOW__COMPANY_PAGE","SHOW__COMPANY_COVERAGE_PAGE","SHOW__COMPANY_ENTITLEMENTS_PAGE"
 Supervisor":{"can":
["FILTER__ENTITLEMENTS","SHOW__EMPLOYEE","SHOW__FILTER_OPTIONS","SHOW__SUPERVISOR_PAGE","SHOW__SUPERVISOR_DETAI
 User":{"can":["SHOW__CERTIFICATIONS","SEARCH__USER","SHOW__ENTITLEMENT","SHOW__USER"]}}}'
```

Chapter 3
Authentication

The following are Autonomous Identity authentication endpoints:

POST Login

Log in to the system. The endpoint accepts the `username` and `password` in the body of the request. The token provided has an expiry date that can be obtained by decoding the returned JWT and using the `exp` data inside the token.

Endpoint

```
/api/authentication/login
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "username": "admin@test.com",
 "password": "test"
}
```

Example Request

```
curl --location --request POST '/api/authentication/login' \
--header 'Content-Type: application/json' \
--data-raw '{
 "username": "admin@test.com",
 "password": "test"
}'
```

POST renewToken

Renew a token for the system. The endpoint accepts the JWT in the header `Authorization: Bearer $JST`. The expiry time of the token is reset and return in the new token.

Endpoint

```
/api/authentication/renewToken
```

Authorization

```
Token             {{token}}
```

Headers

```
Content-Type      application/json
```

Body

```
''
```

Example Request

```
curl --location --request POST '/api/authentication/renewToken' \
--header 'Content-Type: application/json' \
--data-raw ''
```

GET actions

Retrieve the permitted actions of the currently authenticated user.

Endpoint

```
/api/authentication/action
```

Authorization

```
Token            {{token}}
```

Headers

```
Content-Type      application/json
```

Example Request

```
curl --location --request GET '/api/authentication/actions' \
--header 'Content-Type: application/json'
```

Chapter 4 User Details

The following are Autonomous Identity user details endpoints:

POST /

Get user details.

Endpoint

```
/api/userDetails
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "userId": "elizabeth.saiz"
}
```

Example Request

```
curl --location --request POST '/api/userDetails' \
--header 'Content-Type: application/json' \
--data-raw '{
 "userId": "elizabeth.saiz"
}'
```

POST drivingFactor

Get driving factors

Endpoint

```
/api/userDetails/drivingFactor
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "entitlement": "Web_NAS_Share_Case Management_7HQ"
}
```

Example Request

```
curl --location --request POST '/api/userDetails/drivingFactor' \
--header 'Content-Type: application/json' \
--data-raw '{
 "entitlement": "Web_NAS_Share_Case Management_7HQ"
}'
```

POST search

Search for user details.

Endpoint

```
/api/userDetails/search
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "username": "a"
}
```

Example Request

```
curl --location --request POST '/api/userDetails/search' \
--header 'Content-Type: application/json' \
--data-raw '{
 "username": "elizabeth saiz"
}'
```

Example Response

```
{
  "values": [
    {
      "user": "elizabeth.saiz",
      "isentitlementowner": "true",
      "issupervisor": "true",
      "userdisplayname": "Elizabeth Saiz",
      "userdisplayname_lowercase": null
    }
  ]
}
```

POST Entitlements

Search for entitlements.

Endpoint

```
/api/userDetails/search/ent
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "entitlement": "test"
}
```

Example Request

```
curl --location --request POST '/api/userDetails/search/ent' \
--header 'Content-Type: application/json' \
--data-raw '{
 "entitlement": "test"
}'
```

POST Auto Provision

Get auto provision.

Endpoint

```
/api/userDetails/ent/autoprovision
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Body

```
{
 "user": "test"
}
```

Example Request

```
curl --location --request POST '/api/userDetails/ent/autoprovision' \
--header 'Content-Type: application/json' \
--data-raw '{
 "user": "test"
}'
```

Chapter 5
Config

The following are Autonomous Identity config endpoint:

GET /

Get the configuration. This endpoint is mainly used by the Autonomous Identity UI microservice to get values stored in Consul.

Endpoint

```
/api/config
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Example Request

```
curl --location --request GET '/api/config' \
--header 'Content-Type: application/json'
```

Example Response

```
{
  "thresholds": {
    "top": 1.01,
    "high": 0.75,
    "medium": 0.35,
    "low": 0
  },
  "volumeThresholds": {
    "high": 90,
    "low": 20
  }
}
```

Chapter 6
Report

Autonomous Identity captures information in its log files that are useful when troubleshooting problems. You can access the reports using REST calls to the Reports API endpoint.

POST /

Get the configuration. You can specify `outputType` as:

```
outputType: 'csv'
outputType: 'json'
```

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

POST /EventBasedCertification

Get the event based certification report.

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Params

```
fields
```

Body

```
{
 "fields": [
  "id",
  "type",
  "batch_id",
  "original",
  "update"
 ],
 "reportType": "EventBasedCertification"
}
```

Example Request

```
curl --location --request POST '/api/report' \
--header 'Content-Type: application/json' \
--data-raw '{
 "fields": [
  "id",
  "type",
  "batch_id",
  "original",
  "update"
 ],
 "reportType": "EventBasedCertification"
}'
```

POST /RoleMining

Get the role mining report.

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Params

```
fields
```

Body

```
{
 "fields": [
  "entt_id",
  "entt_name",
  "policy",
  "role",
  "total_employees",
  "total_entts"
 ],
 "reportType": "RoleMining"
}
```

Example Request

```
curl --location --request POST '/api/report' \
--header 'Content-Type: application/json' \
--data-raw '{
 "fields": [
  "user_name"
 ],
 "reportType": "RoleMining"
}'
```

POST /AnomalyReport

Get the anomaly report.

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Params

```
fields
```

Body

```
{
 "fields": [
  "app_name",
  "avg_conf_score",
  "confidence",
  "entitlement",
  "entitlement_name",
  "freq",
  "frequnion",
  "justification",
  "last_usage",
  "manager_name",
  "median",
  "num_below_conf_threshold",
  "percent_below_threshold",
  "total_assignees",
  "user",
  "user_name"
 ],
 "reportType": "AnomalyReport"
}
```

Example Request

```
curl --location --request POST '/api/report' \
--header 'Content-Type: application/json' \
--data-raw '{
 "fields": [
  "app_name",
  "avg_conf_score",
  "confidence",
  "entitlement",
  "entitlement_name",
  "freq",
  "frequnion",
  "justification",
  "last_usage",
  "manager_name",
  "median",
  "num_below_conf_threshold",
  "percent_below_threshold",
  "total_assignees",
  "user",
  "user_name"
 ],
 "reportType": "AnomalyReport"
}'
```

POST /RecommendPredictions

Get the Recommend Predictions report.

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Params

```
fields
```

Body

```
{
 "fields": [
  "conf",
  "ent",
  "freq",
  "frequnion",
  "rule",
  "usr_key"
 ],
 "reportType": "RecommendPredictions"
}
```

Example Request

```
curl --location --request POST '/api/report' \
--header 'Content-Type: application/json' \
--data-raw '{
 "fields": [
  "conf",
  "ent",
  "freq",
  "frequnion",
  "rule",
  "usr_key"
 ],
 "reportType": "RecommendPredictions"
}'
```

POST /AutoRecertificationFeed

Get the Auto Recertification Feed report.

Endpoint

```
/api/report
```

Authorization

```
Bearer Token <JWT-value>
```

Headers

```
Content-Type      application/json
```

Params

```
fields
```

Body

```
{
 "fields": [
  "app_id",
  "app_name",
  "auto_recert",
  "chiefyesno",
  "city",
  "costcenter",
  "ent_size",
  "entitlement",
  "entitlement_name",
  "event_recert",
  "freq",
  "frequnion",
  "jobcodename",
  "justification",
  "lineofbusiness",
  "lineofbusinesssubgroup",
  "managername",
  "score",
  "user",
  "user_name",
  "userdepartmentname",
  "userdisplayname",
  "usremptype",
  "usrmanagerkey"
 ],
 "reportType": "AutomaticRecertificationFeed"
}
```

Example Request

```
curl --location --request POST '/api/report' \
--header 'Content-Type: application/json' \
--data-raw '{
 "fields": [
  "app_id",
  "app_name",
  "auto_recert",
  "chiefyesno",
  "city",
  "costcenter",
  "ent_size",
  "entitlement",
  "entitlement_name",
  "event_recert",
  "freq",
  "frequnion",
  "jobcodename",
  "justification",
  "lineofbusiness",
  "lineofbusinesssubgroup",
  "managername",
  "score",
  "user",
  "user_name",
  "userdepartmentname",
  "userdisplayname",
  "usremptype",
  "usrmanagerkey"
 ],
 "reportType": "AutomaticRecertificationFeed"
}'
```

Chapter 7
Company View

The following are Autonomous Identity company view endpoints:

GET /

Get the data for company view.

Endpoint

```
/api/companyview
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview'
```

GET allEntitlementsAvgGroups

Get the company view all entitlements average groups.

Endpoint

```
/api/companyview/allEntitlementAvgGroups
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/allEntitlementAvgGroups'
```

GET entitlementAvgGroupDetails

Get the company view entitlements average groups.

Endpoint

```
/api/companyview/entitlementAvgGroupDetails/0.1/0.15
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/entitlementAvgGroupDetails/0.1/0.15'
```

GET mostCriticalEntitlements

Get the company view most critical entitlements.

Endpoint

```
/api/companyview/mostCriticalEntitlements
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/mostCriticalEntitlements'
```

GET Assignment Stats

Get the company view assignment statistics.

Endpoint

```
/api/companyview/assignmentsStats
```

Authorization

```
Bearer Token <JWT-value>
```

Params

```
assignmentLimit  1
highVolumeHighMinScore  0.9
highVolumentHighMinUsersCount 100
highVolumenLowMaxScore  0.2
highVolumeLowMinUsersCount 100
```

Example Request

```
curl --location --request GET '/api/companyview/assignmentsStats?
assignmentsLimit=1&highVolumeHighMinScore=0.9&highVolumeHighMinUsersCount=100&highVolumeLowMaxScore=0.2&highVol
```

GET assignmentHistConfSummary

Get the company view assignment history summary.

Endpoint

```
/api/companyview/assignmentsHistConfSummary/2020/01
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/assignmentsHistConfSummary/2020/01'
```

GET assignmentHighThreshold

Get the company view assignment high thresholds.

Endpoint

```
/api/companyview/assignments
```

Authorization

```
Bearer Token <JWT-value>
```

Params

```
lowThreshold   true
highThreshold  true
unscored       true
```

Example Request

```
curl --location --request GET '/api/companyview/assignments'
```

GET entitlementsWithoutOwner

Get the company view assignment high thresholds.

Endpoint

```
/api/companyview/entitlementsWithoutOwner
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api/companyview/entitlementsWIthoutOwner'
```

GET usersWithoutManager

Get the company view users without a manager.

Endpoint

```
/api/companyview/usersWithoutManager
```

Authorization

```
Bearer Token <JWT-value>
```

Params

```
lowThreshold   true
highThreshold  true
unscored       true
```

Example Request

```
curl --location --request GET '/api/companyview/usersWithoutManager'
```

GET coverage

Get the company view coverage.

Endpoint

```
/api/companyview/coverage
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/coverage'
```

GET companyViewEntitlementse

Get the company view entitlements.

Endpoint

```
/api/companyview/companyViewEntitlements
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/companyViewEntitlements'
```

GET companyViewEmployeeTypes

Get the company view employee types.

Endpoint

```
/api/companyview/companyViewEmployeeTypes
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api//companyview/companyViewEmployeeTypes'
```

GET entitlementsWithoutOwner

Get the entitlements without an owner.

Endpoint

```
/api/companyview/entitlementsWithoutOwner
```

Authorization

```
Bearer Token <JWT-value>
```

Example Request

```
curl --location --request GET '/api/companyview/entitlementsWIthoutOwner'
```

Chapter 8
Access Control

The following are Autonomous Identity access control endpoints:

POST /

Get access control decision data for actioned user entitlements.

Endpoint

```
/api/accessControl
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "outputType": "csv"
}
```

Example Request

```
curl --location --request POST '/api/accessControl' \
--header 'Content-Type: application/json' \
--data-raw '{
 "outputType": "csv"
}'
```

Example Response

```
i{
    "data": [
        {
            "is_processed": false,
            "entitlement": "tildeNon-Union_Construction_II",
            "user": "george.perez",
            "date_created": "2020-06-16T05:00:22.133Z",
            "role_owner_decision": -1,
            "manager_decision": 1,
            "is_archived": false,
            "manager": "bob.rodgers",
            "manager_date_created": "2020-06-19T07:51:05.533Z",
            "role_owner": "-1",
            "role_owner_auto_certify_reason": null,
            "role_owner_auto_request_reason": null,
            "role_owner_date_created": "1970-01-01T00:00:00.000Z",
            "user_score": null,
            "justification": []
        },
        {
            "is_processed": false,
            "entitlement": "tildeNon-Union_Construction_II",
            "user": "george.perez",
            "date_created": "2020-06-16T05:00:22.133Z",
            "role_owner_decision": -1,
            "manager_decision": 2,
            "is_archived": false,
            "manager": "bob.rodgers",
            "manager_date_created": "2020-06-19T07:50:52.424Z",
            "role_owner": "-1",
            "role_owner_auto_certify_reason": null,
            "role_owner_auto_request_reason": null,
            "role_owner_date_created": "1970-01-01T00:00:00.000Z",
            "user_score": null,
            "justification": []
        }
    ],
    "count": 2
}
```

POST /(get auto certification data)

Get the auto certification data.

Endpoint

```
/api/accessControl
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{"get_auto_certify": true, "get_auto_request": true}
```

Example Request

```
curl --location --request POST '/api/accessControl' \
--header 'Content-Type: application/json' \
--data-raw '{"get_auto_certify": true, "get_auto_request": true}'
```

POST revokeAccess

Revoke access.

Endpoint

```
/api/accessControl/revokeAccess
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "manager": "Christy.Cronin",
 "user": "bloggs",
 "entitlement":"11dbc180-5b86-11e8-957d-37bffaeb9f3a"
}
```

Example Request

```
curl --location --request POST '/api//accessControl/revokeAccess' \
--header 'Content-Type: application/json' \
--data-raw '{
 "manager": "Christy.Cronin",
 "user": "bloggs",
 "entitlement":"11dbc180-5b86-11e8-957d-37bffaeb9f3a"
}'
```

POST batchCertifyAccess

Batch certify access.

Endpoint

```
/api/accessControl/batchCertifyAccess
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "roleOwner": "david.elliott",
 "entitlement": "030ac799-3a51-4a5d-9c58-3deb02081dd5",
 "users": [
  "1111",
  "34534535"
 ]
}
```

Example Request

```
curl --location --request POST '/api//accessControl/batchCertifyAccess' \
--header 'Content-Type: application/json' \
--data-raw '{
 "roleOwner": "david.elliott",
 "entitlement": "030ac799-3a51-4a5d-9c58-3deb02081dd5",
 "users": [
  "1111",
  "34534535"
 ]
}'
```

POST autoCertifyRequestAccess

Auto-certify request access.

Endpoint

```
/api/accessControl/autoCertifyRequestAccess
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "roleOwner":"admin",
 "is_auto_request":true,
 "auto_request_reason":"string",
 "is_auto_certify":true,
 "auto_certify_reason":"string",
 "entitlement":"11dbc180-5b86-11e8-957d-37bffaeb9f3",
 "users": [
  "alphabetti",
  "george"
 ],
    "justification": [
     "JOBCODE_NAME_Bad Engineer",
     "LINE_OF_BUSINESS_Testing"
    ]
}
```

Example Request

```
curl --location --request POST '/api//accessControl/autoCertifyRequestAccess' \
--header 'Content-Type: application/json' \
--data-raw '{
 "roleOwner":"admin",
 "is_auto_request":true,
 "auto_request_reason":"string",
 "is_auto_certify":true,
 "auto_certify_reason":"string",
 "entitlement":"11dbc180-5b86-11e8-957d-37bffaeb9f3",
 "users": [
  "alphabetti",
  "george"
 ],
    "justification": [
     "JOBCODE_NAME_Bad Engineer",
     "LINE_OF_BUSINESS_Testing"
    ]
}'
```

POST cancelAutoCertifyRequest

Revoke access.

Endpoint

```
/api/accessControl/cancelAutoCertifyRequest
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "roleOwner": "david.elliott",
 "entitlement": "WEB_user_Contrusction Affairs Admin_7HQ",
 "users": [
  "elizabeth.saiz"
 ],
 "is_auto_request": true
}
```

Example Request

```
curl --location --request POST '/api/accessControl/cancelAutoCertifyRequest' \
--header 'Content-Type: application/json' \
--data-raw '{
 "roleOwner": "david.elliott",
 "entitlement": "WEB_user_Contrusction Affairs Admin_7HQ",
 "users": [
  "elizabeth.saiz"
 ],
 "is_auto_request": true
}'
```

Chapter 9
Single View with Application

The following are Autonomous Identity single view with applications endpoints:

POST employees

Get an employee's entitlements and statistics.

Endpoint

```
/api/singleViewWithApp/employees
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "employeeId": "elizabeth.saiz",
 "includeLastAccessed": true,
 "pageSize": 5
}
```

Example Request

```
curl --location --request POST '/api//singleViewWithApp/employees' \
--header 'Content-Type: application/json' \
--data-raw '{
 "employeeId": "elizabeth.saiz",
 "pageSize": 2,
 "lastEntitlementId": "0ff681de-ee83-4ab1-82b5-d1cd754a7e28"
}'
```

Example Response

```
{
  "high": 0,
  "medium": 1,
  "low": 1,
  "avg_score": 0.25,
  "app_name": "",
  "app_id": "",
  "entitlement_name": "",
  "high_risk": null,
  "userEntt": [
    {
      "user": "elizabeth.saiz",
      "entitlement": "192aed21-a7d1-40c3-87a3-9dfa4a3d21f5",
      "app_id": "null",
      "app_name": "test3",
      "entitlement_name": "null",
      "freq": null,
      "frequnion": null,
      "high_risk": "null",
      "justification": [],
      "score": 0.1,
      "user_name": "alpha"
    },
    {
      "user": "elizabeth.saiz",
      "entitlement": "36bad416-d42c-47c2-991e-623aa3833028",
      "app_id": "null",
      "app_name": "test6",
      "entitlement_name": "null",
      "freq": null,
      "frequnion": null,
      "high_risk": "null",
      "justification": [],
      "score": 0.4,
      "user_name": "vce"
    }
  ],
  "user": "elizabeth.saiz",
  "entitlementsCount": 14,
  "entitlementsRemainingCount": 10,
  "lastEntitlementId": "36bad416-d42c-47c2-991e-623aa3833028"
}
```

GET entitlements/:entitlementId

Get an entitlement's statistics and list of assigned users.

Endpoint

```
/api/singleViewWithApp/entitlements/0ac4b36b-20d9-4848-a923-0084a7aa581d
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
pageSize       2
lastUserId     bgs
sortDir        desc
onlyLM         1
```

Example Request

```
curl --location --request GET '/api//singleViewWithApp/entitlements/0ac4b36b-20d9-4848-
a923-0084a7aa581d?pageSize=2' \
--header 'Content-Type: application/json'
```

Example Response

```
{
  "high": 1,
  "medium": 0,
  "low": 1,
  "avg_score": 0.5,
  "app_name": "app14",
  "app_id": "null",
  "entitlement_name": "null",
  "high_risk": "null",
  "enntId": "0ac4b36b-20d9-4848-a923-0084a7aa581d",
  "users": [
    {
      "user": "alphabetti",
      "app_id": "null",
      "freq": null,
      "frequnion": null,
      "justification": [],
      "score": 0.1,
      "user_name": "alpha"
    },
    {
      "user": "bgs",
      "app_id": "null",
      "freq": null,
      "frequnion": null,
      "justification": [],
      "score": 0.9,
      "user_name": "bgs"
    }
  ],
  "usersCount": 12,
  "usersRemainingCount": 10,
  "lastUserId": "bgs"
}
```

Chapter 10
Role Owner with Application Oriented

The following are Autonomous Identity role owner with applications endpoints:

POST unscoredEntitlements

Get unscored entitlements for role owners.

Endpoint

```
/api/roleOwnerWithAppOriented/unscoredEntitlements
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "roleOwnerId": "supervisor"
}
```

Example Request

```
curl --location --request POST '/api//roleOwnerWithAppOriented/unscoredEntitlements' \
--header 'Content-Type: application/json' \
--data-raw '{
 "roleOwnerId": "supervisor"
}'
```

Example Response

```
{
  "high": 0,
  "medium": 1,
  "low": 1,
  "avg_score": 0.25,
  "app_name": "",
  "app_id": "",
  "entitlement_name": "",
  "high_risk": null,
  "userEntt": [
    {
      "user": "elizabeth.saiz",
      "entitlement": "192aed21-a7d1-40c3-87a3-9dfa4a3d21f5",
      "app_id": "null",
      "app_name": "test3",
      "entitlement_name": "null",
      "freq": null,
      "frequnion": null,
      "high_risk": "null",
      "justification": [],
      "score": 0.1,
      "user_name": "alpha"
    },
    {
      "user": "elizabeth.saiz",
      "entitlement": "36bad416-d42c-47c2-991e-623aa3833028",
      "app_id": "null",
      "app_name": "test6",
      "entitlement_name": "null",
      "freq": null,
      "frequnion": null,
      "high_risk": "null",
      "justification": [],
      "score": 0.4,
      "user_name": "vce"
    }
  ],
  "user": "elizabeth.saiz",
  "entitlementsCount": 14,
  "entitlementsRemainingCount": 10,
  "lastEntitlementId": "36bad416-d42c-47c2-991e-623aa3833028"
}
```

POST entownuserdata

Get entitlement owner user data.

Endpoint

```
/api/roleOwnerWithAppOriented/entownuserdata
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "roleOwnerId": "elizabeth.saiz"
}
```

Example Request

```
curl --location --request POST '/api//roleOwnerWithAppOriented/entownuserdata' \
--header 'Content-Type: application/json' \
--data-raw '{
 "roleOwnerId": "26713",
 "onlyLM": "1"
}'
```

Example Response

```
{
  "roleOwner": {
    "roleOwnerId": "26713",
    "total_entitlements": 1,
    "total_subordinates": 1,
    "unscoredEntitlements": 0,
    "scoredEntitlements": 1,
    "entitlementsWithNoUser": 0,
    "entitlements": [
      {
        "app_id": "1",
        "app_name": "1",
        "entitlement": "1",
        "entitlement_name": "1",
        "high_risk": "1",
        "high": 0,
        "medium": 0,
        "low": 1,
        "avg": "0.20"
      }
    ],
    "distinctApps": [
      {
        "app_id": "1",
        "app_name": "1"
      }
    ]
  }
}
```

Chapter 11

Manager with Application Oriented

The following are Autonomous Identity manager with application oriented endpoints:

POST unscoredEntitlements

Get unscored entitlements for managers.

Endpoint

```
/api/managersWithAppOriented/unscoredEntitlements
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "managerId": "Christy.Cronin",
 "pageSize": 2,
    "lastEntitlementId": "test2",
    "sortDir": "desc"
}
```

Example Request

```
curl --location --request POST '/api/managersWithAppOriented/unscoredEntitlements' \
--header 'Content-Type: application/json' \
--data-raw '{
 "managerId": "Christy.Cronin",
 "pageSize": 2,
    "lastEntitlementId": "test2",
    "sortDir": "desc"
}'
```

Example Response

```
{
  "managerId": "Christy.Cronin",
  "users": [
    {
      "userId": "bloggs",
      "entt": [
        {
          "entitlement": "test",
          "entitlement_name": null,
          "user_name": null,
          "app_name": null
        }
      ]
    },
    {
      "userId": "elizabeth.saiz",
      "entt": []
    }
  ],
  "entitlementsCount": 4,
  "entitlementsRemainingCount": 0,
  "lastEntitlementId": "test"
}
```

POST supervisor

Get supervisor info.

Endpoint

```
/api/managersWithAppOriented/supervisor
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "managerId": "Christy.Cronin"
}
```

Example Request

```
curl --location --request POST '/api/managersWithAppOriented/supervisor' \
--header 'Content-Type: application/json' \
--data-raw '{
 "managerId": "Christy.Cronin"
}'
```

POST supervisorEntitlements

Get supervisor entitlements.

Endpoint

```
/api/managersWithAppOriented/supervisorEntitlements
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "managerId": "Christy.Cronin"
}
```

Example Request

```
curl --location --request POST '/api/managersWithAppOriented/supervisorEntitlements' \
--header 'Content-Type: application/json' \
--data-raw '{
 "managerId": "Christy.Cronin"
}'
```

POST supervisorUser

Get supervisor User.

Endpoint

```
/api/managersWithAppOriented/supervisorUser
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
{
 "managerId": "Christy.Cronin"
}
```

Example Request

```
curl --location --request POST '/api/managersWithAppOriented/supervisorUser' \
--header 'Content-Type: application/json' \
--data-raw '{
 "managerId": "Christy.Cronin"
}'
```

Chapter 12
Entitlements

The following are Autonomous Identity filtering by entitlements endpoints:

GET Filters by Entt Owners

Get filters by entitlement owner.

Endpoint

```
/api/entitlements/filters?by=entitlementOwner&ownerId=timothy.slack
```

Authorization

```
Bearer Token <JWT-value>
```

GET Filters by Supervisor

Get filters by supervisors.

Endpoint

```
/api/entitlements/filters?by=supervisor&ownerId=albert.pardini
```

Authorization

```
Bearer Token <JWT-value>
```

Body

```
by      supervisor
ownerId  albert.pardini
```

Example Request

```
curl --location --request GET '/api/entitlements/filters&#63;by=supervisor&amp;ownerId=albert.pardini'
 \
--header 'content-type: application/json'
```

POST Statistics by Entt Owner

Set filters by entitlement owners.

Endpoint

```
/api/entitlements/stats?by=entitlementOwner
```

Authorization

```
Bearer Token <JWT-value>
```

Params

```
by      entitlementOwner
```

Body

```
{
 "ownerId": "timothy.slack",
 "isHighRiskOnly": true,
 "isMediumLowRiskOnly": false,
 "isUserEntitlementsIncluded": true,
 "filters": [{
  "type": "app_id",
  "group": "criticality",
  "value": "Essential"
 }]
}
```

Example Request

```
curl --location --request POST '/api/entitlements/stats&#63;by=entitlementOwner' \
--header 'content-type: application/json' \
--data-raw '{
 "ownerId": "timothy.slack",
 "isHighRiskOnly": true,
 "isMediumLowRiskOnly": false,
 "isUserEntitlementsIncluded": true,
 "filters": [{
  "type": "app_id",
  "group": "criticality",
  "value": "Essential"
 }]
}'
```

POST Statistics by Supervisor

Set statistics by supervisor.

Endpoint

```
/api/entitlements/stats?by=supervisor
```

Authorization

```
Bearer Token <JWT-value>
```

Params

```
by      supervisor
```

Body

```
{
 "ownerId": "albert.pardini",
 "isHighRiskOnly": true,
 "isMediumLowScoreOnly": true,
 "isUserEntitlementsIncluded": true,
 "filters": [{
  "type": "app_id",
  "group": "criticality",
  "value": "Essential"
 }]
}
```

Example Request

```
curl --location --request POST '/api/entitlements/stats&#63;by=supervisor' \
--header 'content-type: application/json' \
--data-raw '{
 "ownerId": "albert.pardini",
 "isHighRiskOnly": true,
 "isMediumLowScoreOnly": true,
 "isUserEntitlementsIncluded": true,
 "filters": [{
  "type": "app_id",
  "group": "criticality",
  "value": "Essential"
 }]
}'
```

Glossary

anomaly report
A report that identifies potential anomalous assignments.

as-is predictions
A process where confidence scores are assigned to the entitlements that users have.

confidence score
A score from a scale from 0 to 100% that indicates the strength of correlation between an assigned entitlement and a user's data profile.

data audit
A pre-analytics process that audits the seven data files to ensure data validity with the client.

data ingestion
A pre-analytics process that pushes the seven .csv files into the Cassandra database. This allows the entire training process to be performed from the database.

data sparsity
A reference to data that has null values. Autonomous Identity requires dense, high quality data with very few null values in the user attributes to get accurate analysis scores.

data validation
A pre-analytics process that tests the data to ensure that the content is correct and complete prior to the training process.

driving factor
An association rule that is a key factor in a high entitlement confidence score. Any rule that exceeds a confidence threshold level (e.g., 75%) is considered a driving factor.

entitlement
An entitlement is a specialized type of `assignment`. A user or device with an entitlement gets access rights to specified resources.

insight report
A report that provides metrics on the rules and predictions generated in the analytics run.

recommendation

A process run after the as-is predictions that assigns confidence scores to all entitlements and recommends entitlements that users do not currently have. If the confidence score meets a threshold, set by the `conf_thresh` property in the configuration file, the entitlement will be recommended to the user in the UI console.

resource

An external system, database, directory server, or other source of identity data to be managed and audited by an identity management system.

REST

Representational State Transfer. A software architecture style for exposing resources, using the technologies and protocols of the World Wide Web. REST describes how distributed data objects, or resources, can be defined and addressed.

stemming

A process that occurs after training that removes similar association rules that exist in a parent-child relationship. If the child meets three criteria, then it will be removed by the system. The criteria are: 1) the child must match the parent; 2) the child (e.g., [San Jose, Finance]) is a superset of the parent rule. (e.g., [Finance]); 3) the child and parent's confidence scores are within a +/- range of each other. The range is set in the configuration file.

training

A multi-step process that generates the association rules with confidence scores for each entitlement. First, Autonomous Identity models the frequent itemsets that appear in the user attributes for each user. Next, Autonomous Identity merges the user attributes with the entitlements that were assigned to the user. It then applies association rules to model the sets of user attributes that result in an entitlement access and calculates confidence scores, based on their frequency of appearances in the dataset.