AIDR-DATA Requirements

|  |  |
| --- | --- |
| REQ No.  1 | Project  AIDR-DATA |

Abstract

| **Rev** | **Date** | **Author** | **Status** |
| --- | --- | --- | --- |
| 1 | 10/01/2015 | Ji Lucas | First draft |
| 2 |  |  |  |
|  |  |  |  |

Distribution List

|  |  |
| --- | --- |
| **Reviewers** | **FYI** |
| Ji Lucas |  |

Table of Contents

1. Introduction 4

2. Related Documents/Links/People 4

3. Glossary 4

4. Scope 4

4.1. Goals and Objectives 4

4.2. In Scope 5

4.3. Out of Scope 5

4.4. Assumptions 5

4.5. Constraints 5

4.6. Dependencies 5

4.7. Risks 5

5. Requirements 5

5.1. RESTful API 6

5.2. Rate Limiting 6

5.3. Operations 6

5.4. Domain Model 7

5.5. Non-functional 7

5.5.1. Security 7

5.5.2. QA/Testing 8

Issues/Questions 8

6. Revision History 8

AIDR-DATA

# Introduction

aidr-data is the module that will handle analysis, dashboard, data repo and any data related final output to clients.

aidr-data should expect to perform the below tasks:

1. Expose any data that is related to dashboard
2. twitter data files.
3. Collection configuration info

# Related Documents/Links/People

References in the text throughout this document appear in square brackets (e.g., [1], [JS]).

| **Reference** | **Document/Link/Person/Application** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Glossary

| **Term** | **Definition** |
| --- | --- |
| AIDR | Artificial Intelligence for Disaster Response : <http://aidr-dev.qcri.org/AIDRFetchManager/>  To use the application, user has to have twitter account |
| MicroMappers | A cloned customized of Pybossa platform for tagging by Digital Humanitarians http://clickers.micromappers.org/ |
| AIDR Collection Data | Twitter data that is collected by AIDR based on configuration. AIDR Collection output |
|  |  |
|  |  |

# Scope

## Goals and Objectives

aidr-data should be able to handle any request/responses of data that is generated via module to support data repo, dashboard and data visualization.

## In Scope

1. restful API access based on api\_key only
2. collection configuration data
3. model info
4. human tag info
5. user Info

## Out of Scope

1. Real-time data push
2. Any UI should be handled as separate project.
3. Machine Tag info

## Assumptions

1. Should provide data that residents of AIDR

## Constraints

1. The phase1 will handle only api based data. It means that no real-time dashboard push will be available.
2. Database driven api

## Dependencies

1. AIDR

## Risks

# Requirements

The following subsections define software requirements. Each requirement is labeled as follows:

* **ID.** The unique identifier for the requirement.
* **Requirement.** A clear and concise description of the requirement.
* **Source.** A cross-reference to the source of the requirement.
* **Cat**egory**.** A classification for the requirement using the selections from Table 1.

Table 1 - Requirement Categories

| **Category** | **Description** |
| --- | --- |
| **M**andatory | Required minimum functionality. |
| **O**ptional | Nice to have and will be implemented if time permits. |
| **F**uture | Will not be implemented now but should be considered for a future enhancement. |
| **D**ropped | Requirement determined to be completely out of scope **after** baseline. |
| **E**xisting | Functionality already present in the software (for documentation purposes). |
| **Op**erational | Will not be addressed in the software (documented for training purposes). This requirement should have a related item in the issue list in order to identify an owner. |

## RESTful API

The RESTful API is located at:

http://{aidr.qcri.org}/aidrdata/api

* It expects and returns JSON.
* Required to have **API-KEY** to authenticate & authorize the operation.
* The returned objects will have a **links** and **link** fields, not included in the model in order to support [Hypermedia as the Engine of Application State](http://en.wikipedia.org/wiki/HATEOAS) (also known as HATEOAS), so you can know which are the relations between objects.
* All objects will return a field **link** which will be the absolute URL for that specific object within the API.
* If the object has some parents, you will find the relations in the **links** list.
* The object link will have a tag **rel** equal to **self**, while the parent objects will be tagged with **parent**.
* The **title** field is used to specify the type of the object.

## Rate Limiting

* Rate Limiting has been enabled for all the API endpoints.
* The rate limiting gives any user
* in the headers the following values to throttle user’s requests without problems:
  + **X-RateLimit-Limit**: the rate limit ceiling for that given request
  + **X-RateLimit-Remaining**: the number of requests left for the 15 minute window
  + **X-RateLimit-Reset**: the remaining window before the rate limit resets in UTC epoch seconds

## Operations

delete, create, push will not be allowed. Only data searchm, get and list will be allowed via RESTful API.

GET http://{aidr.qcri.org}/aidrdata/api/{domain-object}/{id}[?api\_key=API-KEY]

GET http://{aidr.qcri.org}/aidrdata/api/{domain-object|id}[?domain-object-field=value&api\_key=API-KEY

Multiple fields can be used separated by the **&** symbol:

GET http://{aidr.qcri.org}/aidrdata/api/{domain-object|id}[?field1=value&field2=value2&api\_key=API-KEY]

| **ID** | **Requirement** | **Source** | **Cat** |
| --- | --- | --- | --- |
| .1 | API-KEY must be provided |  | M |
| .2 | If the search does not find anything, the server will return an empty JSON list [] |  | M |
| .3 | If an error occurs, the action will return a JSON object like this:  **{**  **"status": "failed",**  **"action": "POST",**  **"target": "collection",**  **"exception\_msg": "type object 'collection' has no attribute 'short\_ame'",**  **"status\_code": 415,**  **"exception\_cls": "AttributeError"**  **}** |  | M |
| .4 | Delete, create, push request should be rejected and, proper message should be sent. |  | M |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Domain Model

aidr-data will have the below domain objects :

* Collection
* Collection\_log
* Collection\_summery\_history
* users
* crisis
* document\_nominal\_label
* task\_answer
* model\_family
* model
* model\_nominal\_label

Any associated or child table(s) or attribute(s) should rendered as needed.

## Non-functional

### Security

| **ID** | **Requirement** | **Source** | **Cat** |
| --- | --- | --- | --- |
| .1 | *api-key per user* |  | M |

### QA/Testing

| **ID** | **Use Case** | **Date/Status** |
| --- | --- | --- |
| .1 |  |  |
| 2. |  |  |

# Issues/Questions

| **Issue #** | **Issue/Resolution Description** | **Date/Status** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Revision History

Changes to the text of this document are indicated by bars in the outside margin adjacent to the affected text.

| **Date** | **Change Description** |
| --- | --- |
| 10/01/2015 | Initial draft. |
|  |  |
|  |  |