Jeremy Harrault

Swordfish

– SAFAPS SIM –

API Specification

Objectives of this document

The purpose of this document is to specify the interface of the SAFAPS SIM API. This is the interface between SAFAPS web service and external systems requesting SAFAPS. This document define the functions offered by the API both from static view and dynamic view. The static view of each function is defined by their inputs and outputs whereas the dynamic view is defined using the sequential view showing the action which need to be performed for each function.

Glossary and Terminology

– A –

API: Application Programming Interface

– R –

REST: Representational State Transfer.

– S –

S&F: Stress and Fatigue

SAFAPS: Stress and Fatigue Audit and Prediction Service

Document Description

|  |  |  |
| --- | --- | --- |
| Title | SAFAPS SIM : API Specification | |
| Creation date | 26/01/2016 | |
| Publication date | 28/01/2016 | |
| Product Owner | Augustin Tataru | taau15md@student.ju.se |
| Authors | Jeremy Harrault | hajr15bp@ju.se |
|  |  |
| Subject | API Specification | |
| Model version | 1.0 | |
| Document version | 1.1 | |

Revisions table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Rev. | Author | Modified Section(s) | Comments |
| 26/01/16 | 1.0 | Jeremy Harrault | All | Define static view for all functions |
| 27/01/16 | 1.1 | Jeremy Harrault | 3. | Add dynamic view for all functions |

Table of Contents

[1. Description of the API 1](#_Toc441663260)

[1.1. REST architecture 1](#_Toc441663261)

[1.2. API keys 2](#_Toc441663262)

[1.2.1. Managers’ API secret keys 2](#_Toc441663263)

[1.2.2. Organizations’ API 2](#_Toc441663264)

[1.2.3. Sending API Keys in a request 2](#_Toc441663265)

[2. Resource View 1](#_Toc441663266)

[3. Implementation 2](#_Toc441663267)

[3.1. Evaluations 2](#_Toc441663268)

[3.1.1. Interface 2](#_Toc441663269)

[3.1.2. Dynamic view 5](#_Toc441663270)

[3.2. Manager management 6](#_Toc441663271)

[3.2.1. Interface 6](#_Toc441663272)

[3.2.2. Dynamic view 7](#_Toc441663273)

[3.3. Invoices 8](#_Toc441663274)

[3.3.1. Interface 8](#_Toc441663275)

[3.3.2. Dynamic view 10](#_Toc441663276)

List of Tables

[Table 1: Explanation on request and response attributes 1](#_Toc441663277)

[Table 2: Resource view for SAFAPS SIM API 1](#_Toc441663278)

[Table 3: Interface for S&F evaluation 4](#_Toc441663279)

[Table 4: Interface for manager management 6](#_Toc441663280)

[Table 5: Interface for invoices 9](#_Toc441663281)

List of Figures

[Figure 1: Composition of HTTP request and response 1](#_Toc441663282)

[Figure 2: Dynamic view for S&F evaluation 5](#_Toc441663283)

[Figure 3: Dynamic view for manager creation 7](#_Toc441663284)

[Figure 3: Dynamic view for invoice generation 10](#_Toc441663285)

# Description of the API

## REST architecture

The SAFAPS SIM API fulfil a RESTful architecture. It is reachable using the HTTP protocol. It means that each function offered by the API can be executed by sending an HTTP request and return an HTTP response.

Below is the basic composition of any HTTP response and request.

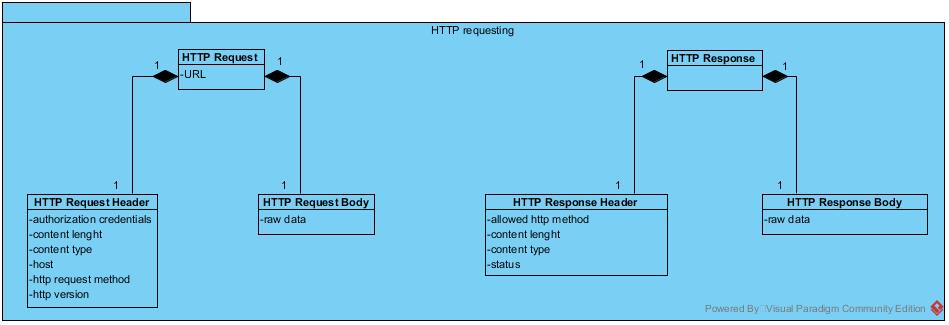


Figure 1: Composition of HTTP request and response

|  |  |  |
| --- | --- | --- |
| Request | Route | This identify the resource on the server to perform an action on |
| Authorization credentials | This field will be used to store the encrypted key allowing the API to authenticate a manager or organization within the system. (cf. 1.2. API keys). |
| Content length | This field contains the number of characters contained in the request body. |
| Content type | This field contains the format of the data sent in the request body. |
| Host | This field contains the host of the requested server |
| HTTP request method | This field specify the action to perform on the route. (GET/POST/PUT/DELETE/etc.) |
| HTTP version | The version the HTTP protocol to be used for the request |
| Raw data | The situational data to send with the request. |
| Response | Allowing HTTP method | The authorized action to perform on the requested resource |
| Content length | This field contains the number of characters contained in the response body. |
| Content type | This field contains the format of the data sent in the response body. |
| Status | This field contains an integer identifying the type of response (OK, redirection, client error, server error). |
| Raw data | The situational data to send back to the caller |

Table 1: Explanation on request and response attributes

For the SAFAPS SIM API, the input and output data in the request and response bodies are formatted in **JSON**.

## API keys

### Managers’ API secret keys

SAFAPS SIM needs to identify the manager who made request to add it to the invoice of his/her organization. To do so, each manager has its own SAFAPS secret key allowing him to be authenticated when he sends a request. Such managers’ API keys are created at the insertion of the managers inside SAFAPS SIM.

### Organizations’ API

Some request offered by SAFAPS SIM’s API cannot be perform by managers but by the organizations themselves. To do so, each organization has a secret key allowing it to be authenticated when sending a request.

### Sending API Keys in a request

The API keys need to be sent in the request header of the requests, in the “Authorization” field. The value of this field is "Basic base64(apiKey)"

# Resource View

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | HTTP Method | API Key | Description |
| /evaluations | POST | Manager | Perform a SAFAPS request to run the S&F algorithm |
| /organizations/managers | POST | Organization | Add a new manager for the organization |
| /organizations/managers/[manid] | DELETE | Organization | Close a manager account for the organization |
| /organizations/invoices | GET | Organization | Retrieve the list of invoices for an organization |
| /organizations/[orgaid]/invoices | POST | NA | Create an invoice for an organization from the performed request |

Table 2: Resource view for SAFAPS SIM API

# Implementation

## Evaluations

### Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request | | | Response | |
| Resource | HTTP Method | Body | HTTP Status | Body |
| /evaluations | POST | *{*  *ResponseURL: “[string]”*  *Schedule: {*  *TimeZone: “[string]”*  *Events: [*  *{*  *StartTime: “[datetime]”*  *EndlTime: “[datetime]”*  *ASMEnvironment: “[enum]”*  *ControlTechnology: “[enum]”*  *ControllerStatus: “[enum]”*  *Traffic: “[enum]”*  *Equipement: “[enum]”*  *Weather: “[enum]”*  *},*  *…*  *]*  *}*  *}*  **Format and possible values:**  TimeZone => “*Continent/City*”  StartTime/EndTime => “*Y-M-d h:m:s*”  ASMEnvironment:   * “*E*” for Enroute * “*T*” for Terminal * “*LM*” for Local Moves * “*D*” for Departures * “*A*” for Arrivals * “*GM*” for Ground Moves * “*T*” for Tower   ControlTechnology:   * “*R*” for Radar * “*PS*” for Procedural – Supported * “PM” for Procedural – Manual   ControllerStatus:   * “*SC*” for Solo controller * “*MCU*” for Multi controller – Unsupported * “*MCS*” for Multi controller – Supported * “*MCM*” for Multi controller – Mentoring * “*MCT*” for Multi controller – Trainee * “*MCI*” for Multi controller – instructor   Traffic:   * “*VH*” for Very Heavy * “*H*” for Heavy * “*B*” for Busy * “*NB*” for Not busy * “*L*” for Light * “*VL*” for Very Light   Equipement:   * “*SD*” for Severely Degraded * “*BD*” for Badly Degreded * “*D*” for Degreded * “*O*” for Operational   Weather:   * “*HD*” for Highly disruptive * “*D*” for Disruptive * “*MD*” for Mildly Disruptive * “*ND*” for No Disruption | 204 | EMPTY |
| 400 | *{*  *“Error”: “The response URL is not properly set”*  *}* |
| 400 | *{*  *“Error”: “The schedule time zone is not properly set”*  *}* |
| 400 | *{*  *“Error”: “One or several event attributes are missing in the event [event\_nbr]: “[ev\_missing\_attr1]”, “[ev\_missing\_attr2]”, …*  *}*  The [event\_nbr] refers to the index (starting at 0) of the uncomplete event within the JSON array contained in the request body.  The [ev\_missing\_attrn] refers to the name of the missing attribute. |
| 401 | *{*  *“Error”: “The authorization field is not specified or the API key is not valid”*  *}* |
| 415 | *{*  *“Error”: “The data is not in JSON format”*  *}* |

Table 3: Interface for S&F evaluation

### Dynamic view

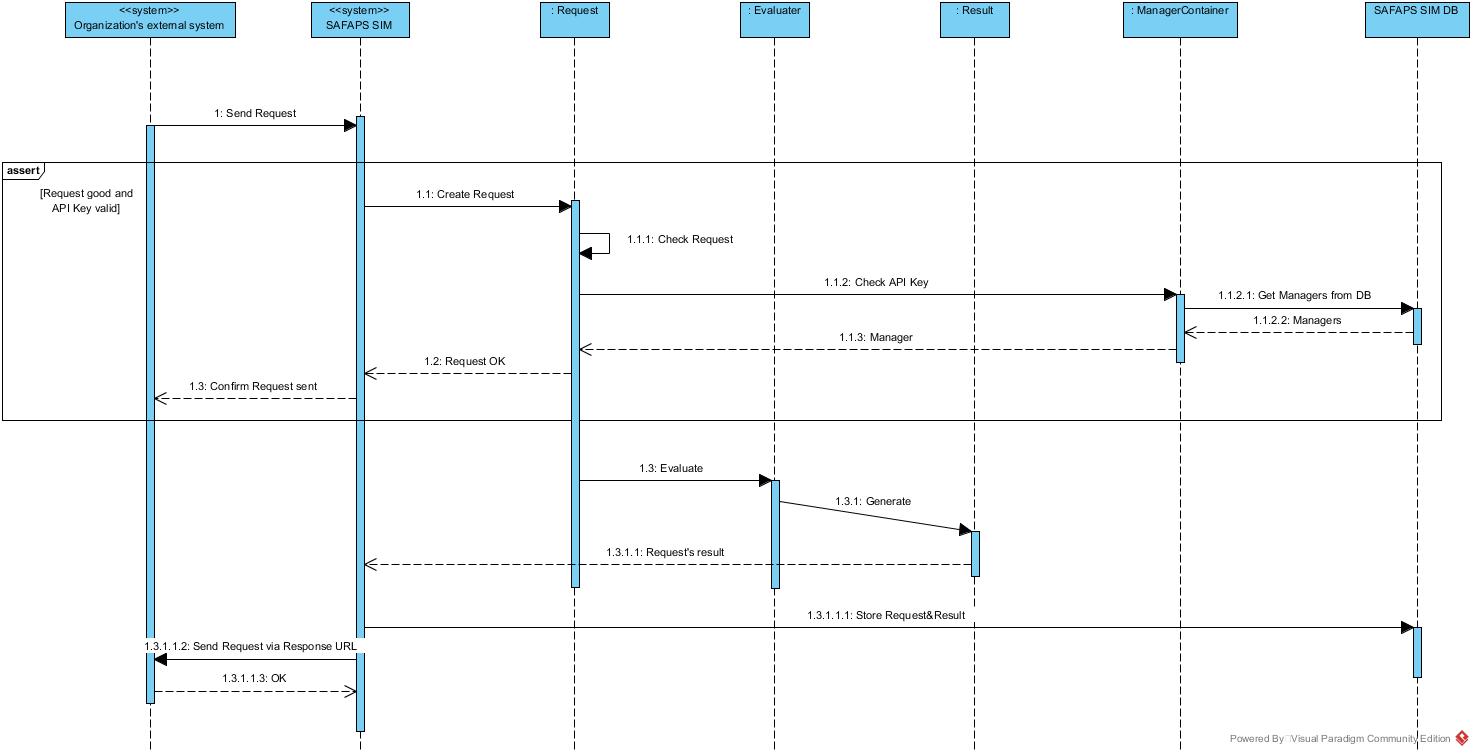


Figure 2: Dynamic view for S&F evaluation

## Manager management

### Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request | | | Response | |
| Resource | HTTP Method | Body | HTTP Status | Body |
| /organizations/managers | POST | *{*  *Name: “[string]”*  *}* | 200 | *{*  *ManagerId: [integer]*  *Name: “[string]”*  *ApiKey: “[string]”*  *}* |
| 400 | *{*  *“Error”: “The name is not properly set”*  *}* |
| 401 | *{*  *“Error”: “The authorization field is not specified or the API key is not valid”*  *}* |
| 415 | *{*  *“Error”: “The data is not in JSON format”*  *}* |
| /organizations/managers/[manid] | DELETE | EMPTY | 204 | EMPTY |
| 400 | *{*  *“Error”: “The manager id is not valid”*  *}* |
| 401 | *{*  *“Error”: “The authorization field is not specified or the API key is not valid”*  *}* |

Table 4: Interface for manager management

### Dynamic view

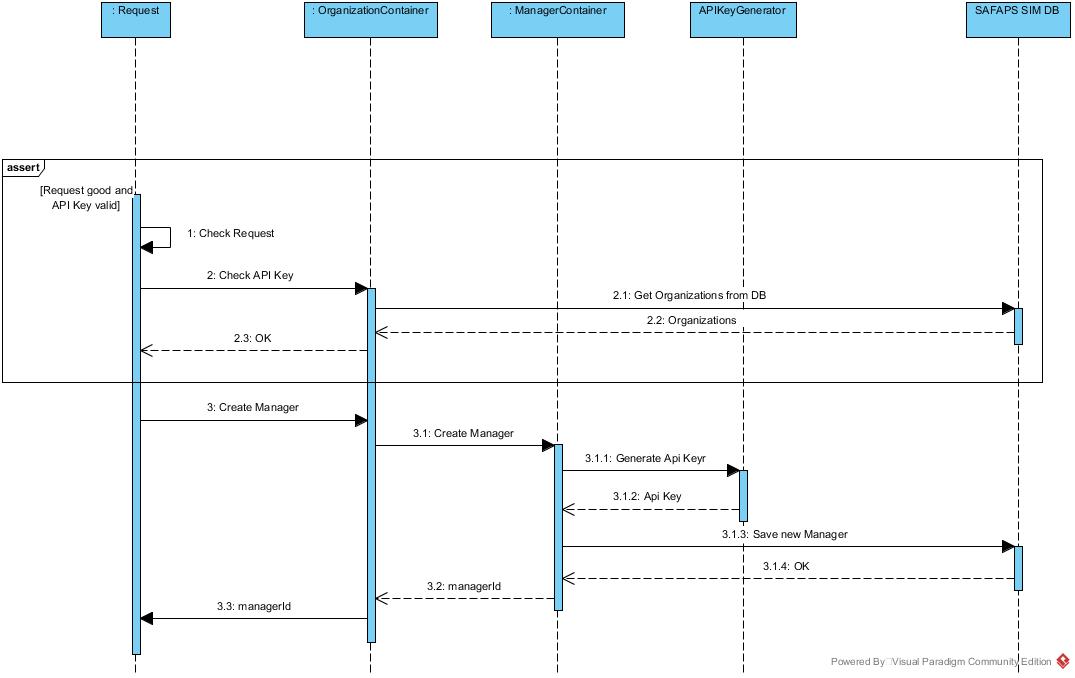


Figure 3: Dynamic view for manager creation

## Invoices

### Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request | | | Response | |
| Resource | HTTP Method | Body | HTTP Status | Body |
| /organizations/[orgaId]/invoices | POST | EMPTY | 200 | *{*  *OrganizationName:”[string]”*  *Invoice: {*  *Date: ”[datetime]”*  *PeriodStart: “[datetime]”*  *PeriodEnd: “[datetime]”*  *Amount: [float]*  *Currency: “[string]”*  *}*  *}*  PeriodStart and PeriodEnd refers to the time frame to consider as invoiced.  Date is the date the invoice is instantiated. |
| 400 | *{*  *“Error”: “The organization id is not valid”*  *}* |
| /organizations/invoices | GET | EMPTY | 204 | {  OrganizationName: “[string]”  Invoices: [  *{*  *Date: ”[datetime]”*  *PeriodStart: “[datetime]”*  *PeriodEnd: “[datetime]”*  *Amount: [float]*  *Currency: “[string]”*  *},*  *…*  ]  } |
| 401 | *{*  *“Error”: “The authorization field is not specified or the API key is not valid”*  *}* |

Table 5: Interface for invoices

### Dynamic view

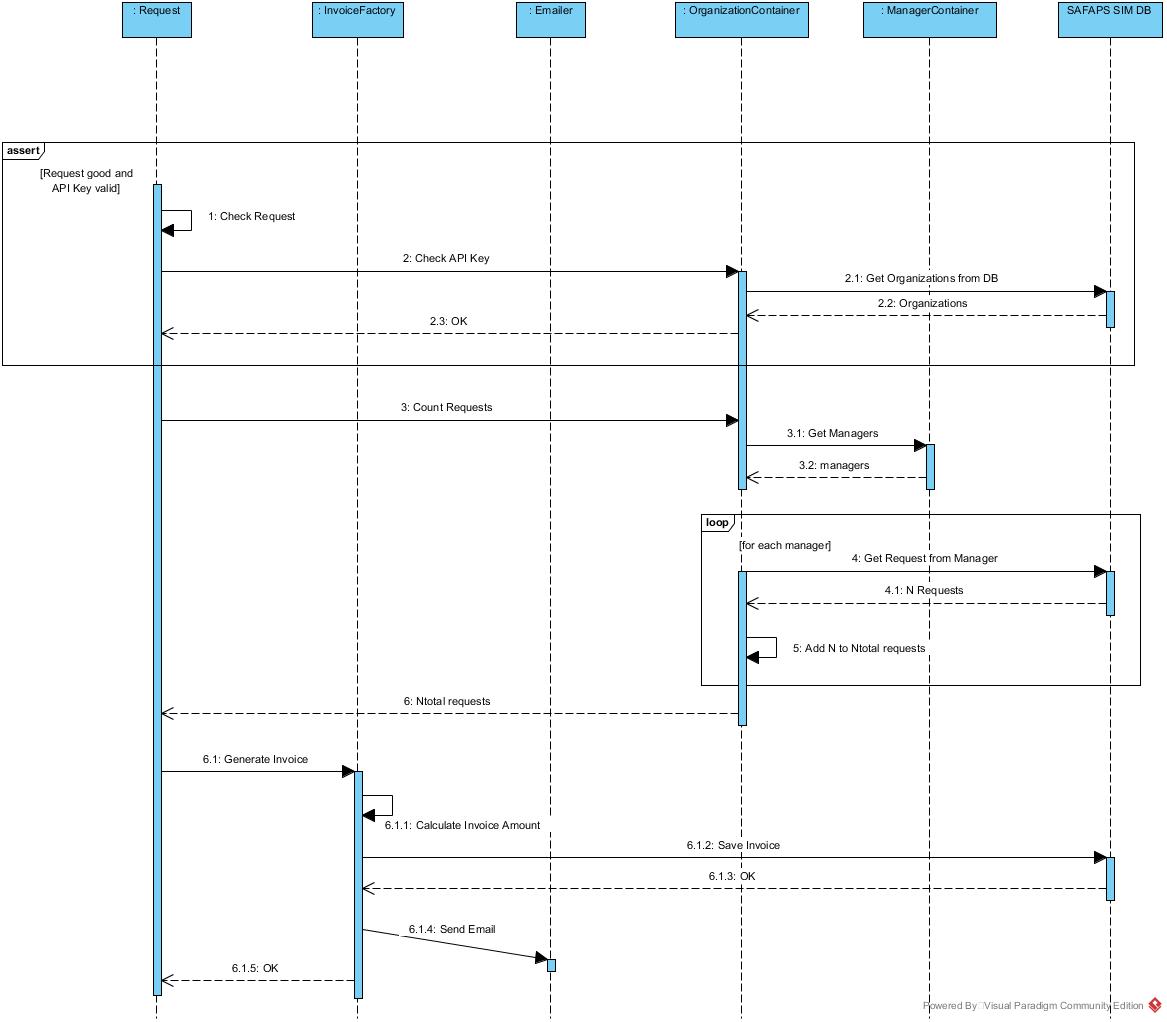


Figure 3: Dynamic view for invoice generation