|  |  |
| --- | --- |
|  | **Web-Engineering I**  **TINF19A** |

Description Web services Calendar

This document contains a short description of the web services provided to manage the calendar. For questions / in case of issues please send an E-Mail to [danny@dxc.com](mailto:danny@dxc.com)

A YAML file describing the REST Web Service is also provided.

# General

* Data management and storage is done by the provided Web services. Task of the exam project is the interactive frontend only.
* The Web services are supporting different accounts. The account name has to be sent with every request (no authentication). Each student / group needs to select a unique account name (e.g. Matrikelnummer)
* The Web services have been developed using PHP, data storage is in a MySQL database.

# Web service URL

* The web service is deployed at URL: <http://dhbw.radicalsimplicity.com/calendar/>
* Note: The web service is sending the HTTP headers required for CORS (Cross-Origin Resource Sharing). Though that might not be supported by all browsers.

# Error Handling

* In case of success, HTTP status code 200 and the respective JSON result will be returned.
* In case of an error, a special result and a HTTP status code different than 200 will be returned:
  + {  
     "error": true,  
     "code": "1",  
     "description": "No action given"  
    }
* Every error contains a unique ID and a brief description

# Web Services

## Calendar Entries

### Retrieve all available events

|  |  |
| --- | --- |
| Description: | This method lists all available events for an account. |
| Resource: | /{user}/events |
| HTTP method: | GET |
| Parameter: | - None - |
| Validations: | - None - |
| HTTP Status: | 200 |
| Result Body: | For each available event, the details are returned Category assignments (if available) will be returned as an array.  [  {  "id": 1,  "title": " Christmas Feast",  "location": "Stuttgart",  "organizer": "dhbw@radicalsimplicity.com",  "start": "2014-12-24T18:00",  "end": "2014-12-24T23:00",  "status": "Busy",  "allday": false,  "webpage": "http://www.radicalsimplicity.com/",  "imageurl": " http://www.radicalsimplicity.com/logo.png",  "categories": [  {  "id": 1,  "name": "Private"  },  {  "id": 5,  "name": "DHBW"  }  ],  "extra": null  },  {  "id": 18,  "title": "New Year’s Eve",  "location": null,  "organizer": "hans@maier.com",  "start": "2014-12-24T18:00",  "end": "2014-12-24T23:30",  "status": "Busy",  "allday": false,  "webpage": "http://www.test.de/",  "imageurl": null,  "categories": [  {  "id": 1,  "name": "Family"  },  {  "id": 3,  "name": "Private"  }  ] ,  "extra": null  }  ] |

Test url:

https://img.favpng.com/14/7/2/pepe-the-frog-portable-network-graphics-pol-jpeg-imgur-png-favpng-ah5eXEAHVTweTDZ6mF7vXeUfQ.jpg

### Create event

|  |  |
| --- | --- |
| Description: | This method creates a new event. |
| Resource: | /{user}/events |
| HTTP method: | POST |
| Parameter: | {  "title": " Christmas Feast",  "location": "Stuttgart",  "organizer": "dhbw@radicalsimplicity.com",  "start": "2014-12-24T18:00",  "end": "2014-12-24T23:30",  "status": "Busy",  "allday": false,  "webpage": " http://www.radicalsimplicity.com/",  "imagedata": "data:*ContentType*;base64,*ImageContent*",  "categories": [  {  "id": 3  }  ],  "extra": null  }  Fields:   * title – title of the new event * location – location of the event * organizer – organizer of the event * start – time when the event starts. (for all day events, this shall be set to 00:00) * end – time when the event ends. (for all day events, this shall be set to the time 23:59) * status – availability status of the event * allday – true/false to define all day events * webpage – webpage for details * imagedata – (Optional) image of the event [see Image Create service for details] If not defined / value is null, no image will be added. If value is set to string value “REMOVE” (case-sensitive), an existing image will be removed. * categories – (Optional) array of categories to which the event shall be assigned to * extra – (Optional) structure to store additional information. Serialized version might not be longer than 2.000 characters.   Supported formats for time information:   * yyyy-MM-dd'T'HH:mm * yyyy-MM-dd'T'HH:mm:ss'Z' * yyyy-MM-dd'T'HH:mm:ss.mmm'Z' |
| Validations: | * title – mandatory, max. 50 characters * location – optional, max. 50 characters * organizer – mandatory, max. 50 characters. Needs to contain a valid E-Mail address * start – mandatory, see format above * end – mandatory, see format above * status – mandatory, one of the values “Free” / “Busy” / “Tentative” * allday – optional, one of the values “true” / ”false”. If not given, it is assumed to be “false” * webpage – optional, max. 100 characters * imagedata – see validations in “Add Image” service * categories – if entries are defined, they need to refer to IDs of valid categories |
| HTTP Status: | 200 |
| Result Body: | {  "title": " Christmas Feast",  "location": "Stuttgart",  "organizer": "dhbw@radicalsimplicity.com",  "start": "2014-12-24T18:00",  "end": "2014-12-24T23:00",  "status": "Busy",  "allday": false,  "webpage": " http://www.radicalsimplicity.com/",  "imageurl": null,  "categories": [],  "extra": null  } |

### Retrieve event

|  |  |
| --- | --- |
| Description: | This method returns an event by ID. |
| Resource: | /{user}/events/{event-id} |
| HTTP method: | GET |
| Parameter: | - None - |
| Validations: | - None - |
| HTTP Status: | 200 |
| Result Body: | {  "id": 1,  "title": " Christmas Feast",  "location": "Stuttgart",  "organizer": "dhbw@bisswanger.de",  "start": "2014-12-24T18:00",  "end": "2014-12-24T23:00",  "status": "Busy",  "allday": false,  "webpage": " http://www.radicalsimplicity.com/",  "imageurl": " http://www.radicalsimplicity.com/logo.png",  "categories": [  {  "id": 1,  "name": "Private"  },  {  "id": 5,  "name": "DHBW"  }  ],  "extra": null  } |

### Update event

|  |  |
| --- | --- |
| Description: | This method updates an existing event.  Note: when updating an event, always all fields have to be transmitted to the server, also thise which have not been changed. |
| Resource: | /{user}/events/{event-id} |
| HTTP method: | PUT |
| Parameter: | * event-id – Unique ID of the selected event (see “list” web service) * All other parameters as defined in “Create event” above |
| Validations: | * The event ID needs to be referencing a valid event for the current account. * Constraints defined in “Create event” apply here as well. |
| HTTP Status: | 200 |
| Result Body: | Same as for Create event |

### Delete event

|  |  |
| --- | --- |
| Description: | This method deleted an existing event. |
| Resource: | /{user}/events/{event-id} |
| HTTP method: | DELETE |
| Parameter: | * id – Unique ID of the selected event (see “list” web service) |
| Validations: | * The event ID needs to be referencing a valid event for the current account. |
| HTTP Status: | 204 |
| Result Body: | - None - |

## Category Management

### List categories

|  |  |
| --- | --- |
| Description: | This method lists all available categories. |
| Resource: | /{user}/categories |
| HTTP method: | GET |
| Parameter: | - None - |
| Validations: | - None - |
| HTTP Status: | 200 |
| Result Body: | [  {  "id": 1,  "name": "Family"  },  {  "id": 3,  "name": "Private"  }  ] |

### Create category

|  |  |
| --- | --- |
| Description: | This method creates a new category. |
| Resource: | /{user}/categories |
| HTTP method: | POST |
| Parameter: | {  "name": "Private"  } |
| Validations: | * name may not contain more than 30 characters |
| HTTP Status: | 200 |
| Result Body: | {  "id": 1,  "name": "Private"  } |

### Retrieve category

|  |  |
| --- | --- |
| Description: | This method returns a category by ID. |
| Resource: | /{user}/categories/{category-id} |
| HTTP method: | GET |
| Parameter: | - None - |
| Validations: | - None - |
| HTTP Status: | 200 |
| Result Body: | {  "id": 1,  "name": "Family"  } |

### Delete category

|  |  |
| --- | --- |
| Description: | This method removes an existing category. |
| Resource: | /{user}/categories/{category-id} |
| HTTP method: | DELETE |
| Parameter: | * category-id – Unique ID of the selected category (see “list-categories” web service) |
| Validations: | * The category ID needs to be referencing a valid category for the current account. |
| HTTP Status: | 204 |
| Result Body: | - None - |

## Category Assignment

### Add event to category

|  |  |
| --- | --- |
| Description: | This method adds a category to an existing event. |
| Resource: | /{user}/categories/{category-id}/{event-id} |
| HTTP method: | POST |
| Parameter: | * event-id – Unique ID of the selected event (see “list” web service) * category-id – Unique ID of the selected category (see “list-categories” web service) |
| Validations: | * The event ID needs to be referencing a valid event for the current account. * The category ID needs to be referencing a valid category for the current account. |
| HTTP Status: | 204 |
| Result Body: | - None - |

### Remove event from category

|  |  |
| --- | --- |
| Description: | This method removes a category from an existing event. |
| Resource: | /{user}/categories/{category-id}/{event-id} |
| HTTP method: | DELETE |
| Parameter: | * event-id – Unique ID of the selected event (see “list” web service) * category-id – Unique ID of the selected category (see “list-categories” web service) |
| Validations: | * The event ID needs to be referencing a valid event for the current account. * The category ID needs to be referencing a valid category for the current account. |
| HTTP Status: | 204 |
| Result Body: | - None - |

## Image Handling

### Add image to event

|  |  |
| --- | --- |
| Description: | This method adds (or updates) an image to an existing event. |
| Resource: | /{user}/images/{event-id} |
| HTTP method: | POST |
| Parameter: | * event-id – Unique ID of the selected event (see “list” web service)   {  "imagedata": "data:*ContentType*;base64,*ImageContent*" } |
| HTTP Status: | 204 |
| Result Body: | - None - |
| Validations: | * The event ID needs to be referencing a valid event for the current account. * The image is in JPEG / PNG format. * The image does not exceed maximum size of 500 kB. |

### Remove image from event

|  |  |
| --- | --- |
| Description: | This method removes an image from an existing event. |
| Resource: | /{user}/images/{event-id} |
| HTTP method: | DELETE |
| Parameter: | * event-id – Unique ID of the selected event (see “list” web service) |
| HTTP Status: | 204 |
| Result Body: | - None - |
| Validations: | * The event ID needs to be referencing a valid event for the current account. * The event needs to have an image assigned. |