1. **Appointment API**
2. **Get**
   * Request

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Url** | | |
| **GET** | api/appointment | | |
| Type | Params | Values | Description |
| HEADER | **Email** | string | Email which is used for accessing server api. |
| HEADER | **Password** | string | Password of email |
| HEADER | **Accept-Language** | string | Language which messages sent from server will be formatted.   * Default : **en-US** |
| URL | **Id** | int | Id of appointment |

* + Response

|  |  |
| --- | --- |
| Status | Content |
| **200** | {  "Appointment": {  "Id": 0,  "Maker": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "Dater": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "From": 0,  "To": 0,  "Note": "",  "Created": 0,  "LastModified": 0,  "Status": 0  }  } |
| **401**  No email or password contained in request header. Front-end should redirect user to login page. | {  "Error": "W012"  } |
| **401**  Account has been disabled. | {  "Error": "W002"  } |
| **401**  Account is pending. | {  "Error": "W001"  } |
| **403**  Account is not allowed to access the function. | {  "Error": "W013"  } |
| **404**  Record is not found. | {  "Error": "W006"  } |

1. **Create**
   * Request

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Url** | | |
| **POST** | api/appointment | | |
| Type | Params | Values | Description |
| HEADER | **Email** | string | Email which is used for accessing server api. |
| HEADER | **Password** | string | Password of email |
| HEADER | **Accept-Language** | string | Language which messages sent from server will be formatted.   * **Default** : **en-US** |
| BODY | **Dater** | int | Id of person that the requester want to make appointment with.   * **Required.** |
| BODY | **From** | double | Time when appointment should be started.   * **Required** |
| BODY | **To** | double | Time when appointment should be ended.   * **Required** |
| BODY | **Note** | string | Note of appointment.   * **Optional.** |

* + Response

|  |  |
| --- | --- |
| Status | Content |
| **200** | {  "Appointment": {  "Id": 0,  "Maker": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "Dater": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "From": 0,  "To": 0,  "Note": "",  "Created": 0,  "Status": 0  }  } |
| **400**  Request parameters are invalid. | {  "Errors": ["", "", ""]  } |
| **401**  No email or password contained in request header. Front-end should redirect user to login page. | {  "Error": "W012"  } |
| **401**  Account has been disabled. | {  "Error": "W002"  } |
| **401**  Account is pending. | {  "Error": "W001"  } |
| **403**  Account is not allowed to access the function. | {  "Error": "W013"  } |
| **403**  Dater is not found, therefore, appointment making is forbidden. | {  "Error": "W008"  } |
| **403**  Dater is not a doctor or patient, therefore, appointment cannot be made. | {  "Error": "W009"  } |
| **403**  Requester and dater have the same role, which means they’re either doctors or patients. Only doctor can date patient and vice versa. | {  "Error": "W011"  } |
| **403**  No relationship between requester and dater exists. | {  "Error": "W019"  } |

1. **Edit**
   * Request

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Url** | | |
| **PUT** | api/appointment | | |
| Type | Params | Values | Description |
| HEADER | **Email** | string | Email which is used for accessing server api. |
| HEADER | **Password** | string | Password of email |
| HEADER | **Accept-Language** | string | Language which messages sent from server will be formatted.   * **Default** : **en-US** |
| URL | **Id** | int | Id of appointment.   * **Required.** |
| BODY | **From** | double | Time when appointment should be started.   * **Optional.** |
| BODY | **To** | double | Time when appointment should be ended.   * **Optional.** |
| BODY | **Note** | string | Note of appointment.   * **Optional.** |
| BODY | **Status** | int | Status of appointment.   * **Optional.** * **0: Cancelled** * **2: Active** * **3: Done** |

* + Response

|  |  |
| --- | --- |
| Status | Content |
| **200** | {  "Appointments": {  "Id": 0,  "Maker": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "Dater": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "From": 0,  "To": 0,  "Note": "",  "Created": 0,  "Status": 0,  "LastModified": 0  }  } |
| **400**  Request parameters are invalid. | {  "Errors": ["", "", ""]  } |
| **401**  No email or password contained in request header. Front-end should redirect user to login page. | {  "Error": "W012"  } |
| **401**  Account has been disabled. | {  "Error": "W002"  } |
| **401**  Account is pending. | {  "Error": "W001"  } |
| **403**  Account is not allowed to access the function. | {  "Error": "W013"  } |
| **403**  Partner is not found or non-active. | {  "Error": "W015"  } |
| **404**  No appointment is found. | {  "Error": "W006"  } |

1. **Filter**
   * Request

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Url** | | |
| **POST** | api/appointment/filter | | |
| Type | Params | Values | Description |
| HEADER | **Email** | string | Email which is used for accessing server api. |
| HEADER | **Password** | string | Password of email |
| HEADER | **Accept-Language** | string | Language which messages sent from server will be formatted.   * **Default** : **en-US** |
| BODY | **Mode** | int | What mode appointment should be filtered.   * **0 : Partner is requester (default)** * **1 : Partner is dater** * **Null : Partner is not required, filter all appointment user takes part in.**   This means, appointments will be filtered by role of partner. If **0** is specified, appointments which have partner as the requester will be shown up and so on. |
| BODY | **Partner** | int | Id of partner   * **Required.** |
| BODY | **MinCreated** | double | Time before which appointment was created.   * **Optional.** * **Year > 1916.** |
| BODY | **MaxCreated** | double | Time after which appointment had been created.   * **Optional.** * **Year > 1916.** |
| BODY | **MinFrom** | double | Time range when appointment should be started.   * **Optional** * **Year > 1916.** * **MinFrom <= MaxFrom** |
| BODY | **MaxFrom** | double |
| BODY | **MinTo** | double | Time range when appointment should be ended.   * **Optional.** * **Year > 1916** * **MinTo <= MaxTo** |
| BODY | **MaxTo** | double |
| BODY | **MinLastModified** | double | Time after when appointment was lastly modified.   * **Optional.** * **Year > 1916** * **<= MaxLastModified** |
| BODY | **MaxLastModified** | double | Time before when appointment had been modified.   * **Optional** * **Year > 1916** * **>= MinLastModified** |
| BODY | **Status** | int | Status of appointment.   * **Optional** * **0: Cancelled** * **1: Pending** * **2: Active** * **3: Done** * **Null : All** |
| BODY | **Page** | int | Index of result page   * **Min : 0 (default)** |
| BODY | **Records** | int | Number of records displayed per page.   * **Min : 1** * **Max : 20 (default)** |

* + Response

|  |  |
| --- | --- |
| Status | Content |
| **200** | {  "Appointments": [  {  "Id": 0,  "Maker": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "Dater": {  "Id": 0,  "FirstName": "",  "LastName": ""  },  "From": 0,  "To": 0,  "Note": "",  "Created": 0,  "Status": 0,  "LastModified": 0  }  ],  "Total": 0  } |
| **400**  Request parameters are invalid. | {  "Errors": ["", "", ""]  } |
| **401**  No email or password contained in request header. Front-end should redirect user to login page. | {  "Error": "W012"  } |
| **401**  Account has been disabled. | {  "Error": "W002"  } |
| **401**  Account is pending. | {  "Error": "W001"  } |
| **403**  Account is not allowed to access the function. | {  "Error": "W013"  } |
| **403**  Dater is not found, therefore, appointment making is forbidden. | {  "Error": "W008"  } |
| **403**  Dater is not a doctor or patient, therefore, appointment cannot be made. | {  "Error": "W009"  } |
| **403**  Requester and dater have the same role, which means they’re either doctors or patients. Only doctor can date patient and vice versa. | {  "Error": "W011"  } |
| **403**  No relationship between requester and dater exists. | {  "Error": "W019"  } |