APIS :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | URI | HTTP Verbs | Description | Request Structure | Response Structure |
| 1 | /v1/persons | POST, GET | POST: Creates a new person  GET: Gets all persons (admin functionality)  Validation Rules while creating a user:   1. First Name shouldn’t be null or empty 2. Last Name shouldn’t be null or empty 3. Email Id shouldn’t be null or empty 4. Email should be valid. | {  "firstName": "Partha",  "lastName": "Sanyal",  "email": "partha.sanyal@email.com"  } | {  "personId": "00a476d3-c73b-4e6a-b8f0-f61eb6dcac09",  "firstName": "Partha",  "lastName": "Sanyal",  "email": "partha.sanyal@email.com",  "\_links": {  "self": {  "href": "http://localhost:8080/v1/persons/00a476d3-c73b-4e6a-b8f0-f61eb6dcac09"  },  "allUsers": {  "href": "http://localhost:8080/v1/persons"  }  }  } |
| 2 | /v1/persons/{person-id} | GET | Gets the details of one user |  | Same as above |
| 3 | /v1/projects | GET | Get all projects functionality has not been implemented as a GET on the base URL, there is a filter implementation for that with filter type = ALL. This API just returns the filter option |  | {  "\_links": {  "filter": {  "href": "http://localhost:8080/v1/projects/filter{?type=ALL/OPEN/CLOSED,projectOwnerId}",  "templated": true  }  }  } |
| 4 | /v1/projects/filter?type=OPEN | GET | Lists all projects that are open for bidding.  Other options for type are “closed”, “all”, anything else for type is invalid.  A second filter parameter can be provided as well “projectOwnerId” – the API will then filter out all projects that have a certain Owner.  Either of the two parameters “type” or “projectOwnerId” is required in order to filter. Both can be present as well. |  | [  {  "projectId": "3324fa18-a9ec-45e6-ae13-7894f14f875a",  "projectName": "project Name",  "projectOwnerId": "cc01f773-96e5-41f3-a0b6-259b0ec40eba",  "requirements": "project requirements",  "maximumBudget": 1000,  "lastDateTime": "2018-02-07T04:53:00Z",  "lowestBidAmount": 0,  "\_links": {  "self": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a"  },  "projectOwner": {  "href": "http://localhost:8080/v1/persons/cc01f773-96e5-41f3-a0b6-259b0ec40eba"  },  "bids": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a/bids"  },  "filterProjects": {  "href": "http://localhost:8080/v1/projects"  },  "winningBid": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a/winningBid"  }  }  }  ] |
| 5 | /v1/projects | POST | Create a new project.  Validations:   1. Project name shouldn’t be blank 2. Project owner Id shouldn’t be blank 3. Project owner should exist 4. Requirements shouldn’t be blank 5. Maximum budget shouldn’t be less than or equal to 0 6. Project last date time for accepting bid shouldn’t be in the past: You can’t create a project in the past. 7. Project last date time should be in a zoned UTC ISO 8601 format. | {  "projectName" : "project Name",  "projectOwnerId" : "cc01f773-96e5-41f3-a0b6-259b0ec40eba",  "requirements" : "project requirements",  "maximumBudget" : 1000,  "lastDateTime" : "2018-02-07T04:53:00Z"  } | {  "projectId": "3324fa18-a9ec-45e6-ae13-7894f14f875a",  "projectName": "project Name",  "projectOwnerId": "cc01f773-96e5-41f3-a0b6-259b0ec40eba",  "requirements": "project requirements",  "maximumBudget": 1000,  "lastDateTime": "2018-02-07T04:53:00Z",  "lowestBidAmount": 0,  "\_links": {  "self": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a"  },  "projectOwner": {  "href": "http://localhost:8080/v1/persons/cc01f773-96e5-41f3-a0b6-259b0ec40eba"  },  "bids": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a/bids"  },  "filterProjects": {  "href": "http://localhost:8080/v1/projects"  },  "winningBid": {  "href": "http://localhost:8080/v1/projects/3324fa18-a9ec-45e6-ae13-7894f14f875a/winningBid"  }  }  } |
| 6 | /v1/projects/{project-id} | GET | Retrieves the project with the given project id |  | Same as above. |
| 7 | /v1/projects/{project-id}/bids | GET, POST | GET : Retrieves all the bids for a particular project. Produces an array of bids  POST : Create a new Bid for a give project (by id).  Validations:   1. Project with the given project id should exist 2. Project should be open for bidding: the last date time for accepting bids for the project should be in the future. 3. The bid amount should be more than 0 and less than the maximum bid set by the project owner. 4. Project owner id shouldn’t be blank 5. Project owner should exist 6. Project owner can’t bid for the same project | {  "bidderId" : "9cf76335-4ee5-43ee-93dc-718a2ce5ddaa",  "bidAmount" : 789  } | {  "bidId": "c8eeb14d-e651-438b-9dbf-1968a6343e81",  "bidderId": "9cf76335-4ee5-43ee-93dc-718a2ce5ddaa",  "bidAmount": 789,  "\_links": {  "self": {  "href": "http://localhost:8080/v1/projects/8400b8f1-185a-4cdb-8595-8b8864f2a3b7/bids/c8eeb14d-e651-438b-9dbf-1968a6343e81"  },  "bidder": {  "href": "http://localhost:8080/v1/persons/9cf76335-4ee5-43ee-93dc-718a2ce5ddaa"  },  "project": {  "href": "http://localhost:8080/v1/projects/8400b8f1-185a-4cdb-8595-8b8864f2a3b7"  }  }  } |
| 8 | /v1/projects/{project-id}/bids/{bid-id} | GET | Retrieve a bid by its id and project id |  | Same as Above |
| 9 | /v1/projects/{project-id}/winningBid | GET | After a project has been closed, the scheduler has run, the winning bid is available to view along with other details. |  | Same as Above |