

HISP Tanzania 2023 Graduate software developer - Online Teaser Question

Introduction:

Welcome to our software developer teaser! As part of the selection process, we would like you to showcase your skills by developing a simple To-do application using the API specifications provided. This exercise will allow you to demonstrate your understanding of software development concepts, API integration, and coding proficiency.

The goal of the task is to create a functional To-do application that allows users to manage their tasks and track their progress. The application should allow users to:

- **Add new To-do item.**
- **View the To-do list.**
- **Updated a To-do item.**
- **Change the status of a To-do item.**
- **Delete a To-do item.**

We will provide you with the API specifications necessary for interacting with the backend, so you can focus on the frontend development and user experience.

This teaser evaluates your ability to design and implement a responsive and user-friendly application. We encourage you to use modern software development technologies and best practices to showcase your skills effectively. You may develop either **web** or **mobile** applications using any of the tech stacks that you are comfortable with while making sure you follow the given API specifications.

Please read through the instructions and specifications carefully before starting the development. We value your creativity and attention to detail, so feel free to add any additional features or enhancements that will enhance the application.

In case of any challenges faced during the development of your application, feel free to contact our team through our official email address: dev@hisptanzania.org

API specifications:

- To-do data model:
To get started with this teaser exercise be sure to use

```
1 interface Todo {  
2   id: string; // unique identifiable for the todo item  
3   title: string; // a todo title  
4   description?: string; // a compulsory description of the todo item  
5   completed: boolean; // status of the todo item, wether completed or not  
6   created: string; // ISO string representation of the date when the todo was created  
7   lastUpdated: string; /// ISO string representation of date when todo was updated  
8 }
```

- Accessing the API:
 - The API will be accessed using the provided API endpoints but using a basic authentication process using the below credentials:

```
1 # Basic Authentication credentials  
2 {  
3   username: admin,  
4   password: district  
5 }
```

- Getting the to-do list:
 - To get all to-do items, use the below API:

```
1 # Getting a new todo item
2
3 # API for getting all todo items
4
5 GET https://dev.hispitz.com/dhis2/api/dataStore/${your-name}?fields=.
```

Things to note:

`${your-name}`: should be your name in camel case or snake case.

The query parameter **fields** indicate the fields from the above To-do interface that are fetched. To fetch specified fields use **fields=title,completed** (to get only title and completed), or for all fields use **fields=.** where "." means all fields.

If your name key does not exist, the server will return a 404-Not found response, hence don't be surprised as this is expected to happen on your first run. Make sure to just handle this in your code. This response can look as below:

```
1 {
2   "pager": {
3     "page": 1,
4     "pageSize": 50
5   },
6   "entries": []
7 }
8
```

If there will be todos existing under your name, the response will be having two keys:

- Pager: This is the description of the pagination of to-dos in your to-do list
- Entries: This is a list of paginated of your to-do items

Below is an example of the response with some todos.

```

1  {
2    "pager": {
3      "page": 1,
4      "pageSize": 50
5    },
6    "entries": [
7      {
8        "key": "todo-1",
9        "value": {
10         "id": "todo-1",
11         "title": "Create online teaser",
12         "created": "2023-05-29T07:25:38.128Z",
13         "completed": false,
14         "description": "Make sure the teaser is fun",
15         "lastUpdated": "2023-05-29T07:25:38.128Z"
16       }
17     }
18   ]
19 }

```


NOTE: If you wish you can extend this implementation by adding pagination using query parameters **pageSize** and **page** as in the below example:

```

1  # Getting all todo items in pagination
2
3  # API for getting all todo items with pagination
4
5  GET https://dev.hisptz.com/dhis2/api/dataStore/${your-name}?fields=.&page=1&page=2


```

- Adding a to-do item:
 - To add a new to-do, use the following API example while following the above-mentioned interface for the To-do.




```
1 # Creating a new todo item
2 # API for creating a new todo item
3 POST https://dev.hispitz.com/dhis2/api/dataStore/${your-name}/${todo-id}
4
5 # body object
6 {
7   "id" : ${todo-id},
8   "title": ${todo-title},
9   "description" : ${todo-description},
10  "completed": ${todo-completed},
11  "created": ${todo-created-date},
12  "lastUpdated": ${todo-last-updated}
13 }
```

- The successful response to the above request will be as follows:




```
1 # Todo created success response
2 {
3   "httpStatus": "OK",
4   "httpStatusCode": 200,
5   "status": "OK",
6   "message": "Key created: '${todo-id}'"
7 }
```

- Getting a specific to-do:
 - To get a specific to-do, use the following API request:




```
1 # API for getting a todo item
2 GET https://dev.hispitz.com/dhis2/api/dataStore/${your-name}/${todo-id}
```

- The success response body for getting one to-do looks like this:



```
1 {
2   "id": "todo-1",
3   "title": "Create online teaser",
4   "created": "2023-05-29T07:25:38.128Z",
5   "completed": false,
6   "description": "Make sure the teaser is fun",
7   "lastUpdated": "2023-05-29T07:25:38.128Z"
8 }
```

- If the specified to-do id does not exist, the response will be:



```
1 # Todo item not found response
2 {
3   "httpStatus": "Not Found",
4   "httpStatusCode": 404,
5   "status": "ERROR",
6   "message": "Key '${your-name}' not found in namespace '${todo-id}'",
7   "errorCode": "E1005"
8 }
```

- Updating a to-do:
 - To update a specific to-do item, use the following API:

```
1 # Updating a todo
2 # API for Updating a todo item
3 PUT https://dev.hispitz.com/dhis2/api/dataStore/${your-name}/${todo-id}
4
5 # body object
6 {
7   "id" : ${todo-id},
8   "title": ${todo-title},
9   "description" : ${todo-description},
10  "completed": ${todo-completed},
11  "created": ${todo-created-date},
12  "lastUpdated": ${todo-last-updated}
13 }
```

- The success response will be as follows:

```
1 # Todo created success response
2 {
3   "httpStatus": "OK",
4   "httpStatusCode": 200,
5   "status": "OK",
6   "message": "Key updated: '${todo-id}'"
7 }
```

- This API can as well be used to change the status of the to-do. It should be noted that the new to-do will have contents similar to the passed object.

- Deleting a to-do:
 - To delete a to-do item, use the following API:

```
1  # Deleting a todo
2
3  # API for Deleting a todo item
4
5  DELETE https://dev.hisptz.com/dhis2/api/dataStore/${your-name}/${todo-id}
6
```

- The delete API response will be as follows:

```
1  # Todo deleted success response
2  {
3      "httpStatus": "OK",
4      "httpStatusCode": 200,
5      "status": "OK",
6      "message": "Key '${todo-id}' deleted from namespace '${your-name}'"
7  }
```

Submission instructions:

Before you begin the development process, we want to emphasize that plagiarism will not be tolerated. This teaser is designed to assess your individual skills and abilities as a software developer. Any form of plagiarism, including copying code or utilizing pre-existing solutions, will lead to immediate disqualification from the selection process. We encourage you to showcase your originality, creativity, and problem-solving capabilities. If you have any queries or need clarification on this matter, please feel free to reach out to us.

You can submit your nice simple application on <https://codepen.io/> or GitHub (public repository) and share the link with us (send to dev@hisptanzania.org).

Good luck, and we look forward to reviewing your impressive work.

