fieldbuzz

Assignment for

Junior Software Engineer Level I or II position

1. **Introduction**

This assignment is designed to gauge your skills and give us an idea of how you approach tasks relevant to applying role. It would be great if you could send this over to us within 7 days' time upon receiving the assignment.

2. **Assignment Statement**

You have to develop **an application including user interface (**detail in **Section 4)** which will be used by yourself as a user to submit your basic information and CV file **following provided APIs and restrictions** to our designated Field Buzz server (Read through **Section 3** for API documentation). And your codebase will have to be available in GitHub. Further detail has been stated in **Section 5**.

For backend applicants: Application needs to be developed in Django (a python-based web-framework)

For android applicants: Application needs to be developed preferably in JAVA (Kotlin is still acceptable)

3. API Specification

This assignment involves 3 key APIs. Detail of those APIs can be found below:

3.1. Authentication Token API:

3.1.1. This API specifically provides a AUTH TOKEN. This TOKEN have to be used for all other API calls to properly authenticating into the system and submit your desired data.

3.1.2. API Detail:

3.1.2.1. API URL: https://recruitment.fisdev.com/api/login/

3.1.2.2. Request Type: POST

3.1.2.3. API Payload format: An APPLICATION/JSON type RAW BODY as below:

```
{
    "username": "<provided username in email>",
    "password": "<provided password in email>"
}
```

3.1.2.4. API Payload Field Definitions:

3.1.2.4.1. username: Use the provided username through email inside

3.1.2.4.2. password: Use the provided password through email

3.1.2.5. API Response Detail:

3.1.2.5.1. Token is returned in "token" field in responded JSON object. This token will be used to authenticate further APIs.

3.1.2.6. **Note**: Ignore all the other fields in response JSON.

3.2. Recruitment Information POST API:

This API allows you to submit your application information as per described below.

3.2.1. API Detail:

3.2.1.1. **Test version URL**: Use this URL while you are testing your application.

3.2.1.1.1.1. https://recruitment.fisdev.com/api/v0/recruiting-entities/

3.2.1.2. **Final version submitting URL**: Use this URL in your final version of the application to submit real data. https://recruitment.fisdev.com/api/v1/recruiting-entities/

3.2.1.3. Request Type: POST

3.2.1.4. API Payload format: An APPLICATION/JSON type RAW BODY. Check below example:

```
{
  "tsync_id": "de4b6998-102f-4be4-8c3c-37f5e944a88c",
  "name": "Candidate One",
  "email": "test.mail@example.com",
  "phone": "+880155000000",
  "full_address": "Sample Address, Sample District, Sample Division",
  "name_of_university": "University One",
  "graduation_year": 2016,
  "cgpa": 3.5,
  "experience_in_months": 12,
  "current_work_place_name": "Sample Current Workplace Name",
  "applying_in": "Backend",
  "expected_salary": 30000,
  "field_buzz_reference": "Sample Field Buzz reference name if ANY",
  "github_project_url": "https://github.com/<username>/<project-name>",
  "cv_file": {
    "tsync_id": "be96f0e188a6de73b186c043-7365-4605-85eb-76021f16803f"
  },
  "on_spot_update_time": 1605644298704,
  "on_spot_creation_time": 1605644298702
}
```

3.2.1.5. **API Payload Field Definitions**:

Field	Explanation	Restrictions to handle
tsync_id	UUID type unique string ID for this entity - Needs to be Internally Generated from API Client side	Plain Text, max length = 55, REQUIRED
name	Applicant's name	Plain Text, max length = 256, REQUIRED
email	Applicant's personal mail	Email Address validated Text, max length = 256, REQUIRED
phone	Applicant's phone in local or international format	Plain Text, max length = 14, REQUIRED
full_address	Applicant's full address	Plain Text, max length = 512
name_of_university	Applicant's varsity name	Plain Text, max length = 256, REQUIRED
graduation_year	Applicant's graduation year	Number, max = 2020, min = 2015, REQUIRED
cgpa	Applicant's graduation yea	Number, max = 4.0, min = 2.0
experience_in_months	Professional working experience in months	Number, max = 100, min = 0
current_work_place_name	Current workplace name	Plain Text, max length = 256
applying_in	Applying department, Mobile or Backend	Choice field: Mobile/Backend, REQUIRED
expected_salary	Expected salary	Number, max = 60000, min = 15000, REQUIRED
field_buzz_reference	Refeence name from Field Buzz if any	Plain Text, max lenght = 256
github_project_url	Submit your developed application's github URL	Plain URL type Text, max_length=512, REQUIRED
cv_file	A json object with only Unique UUID field. Upon POST, server will respond with a FILE ID TOKEN in response's cv_file json object (Explained detail in next section).	JSON Object, REQUIRED
cv_file.tsync_id	UUID type unique string ID for cv_file entity - Needs to be internally generated from API Client side	Plain Text, max length = 55, REQUIRED

on_spot_update_time	Update the value when this entry has been updated	Unix Timestamp in Milliseconds
on_spot_creation_time	Set only when the entry has been created for the first time	Unix Timestamp in Milliseconds

3.2.2.Important information on Payload:

3.2.2.1.1. To update an entity, if you POST the same tsync_id with updated information, entity will be updated in server with new information instead of creating a fresh new entity. Here, tsync_id acts like the unique identifier.

3.2.3.Important Note on API Response:

3.2.3.1. If response JSON object is received properly, it means all data has been sent to submitted to server. Now next step is to understand how to submit the actual CV File through next API.

3.2.3.2. In case for the field `cv_file` json object from submitted payload, server will return a JSON Object including **FILE TOKEN ID**. And this `FILE TOKEN ID` will be used to PUT the actual CV FILE using the `CV File Upload API`

3.2.3.3. Allocated file object looks like below format. Note: We have marked field with "//Ignore" which are not required to be part of your concern. Focus on the explained 2 fields only.

```
{
 "tsync_id": "ee4b6998-102f-4be4-8c3c-37f5e944a88c",
 "name": "Candidate One",
 "email": "test.mail@example.com",
 "phone": "+880155000000",
 "full_address": "Sample Address, Sample District, Sample Division, Bangladesh",
 "name_of_university": "University One",
 "graduation_year": 2016,
 "cgpa": 3.5,
 "experience_in_months": 12,
 "current_work_place_name": "Sample Current Workplace Name",
 "applying_in": "Backend",
 "expected_salary": 30000,
 "field_buzz_reference": "Sample Field Buzz reference name if ANY",
 "cv_file": {
   "id": 1, ---> FILE TOKEN ID
   "tsync_id": "be96f0e188a6de73b186c043-7365-4605-85eb-76021f16803f",
   "code": "FO-00001",// Ignore
   "name": "File_OFOSDYVU_18112020_022552.", // Ignore
   "path": "", // Ignore
   "extension": null, // Ignore
   "description": null, // Ignore
   "file": " // Ignore,
   "date_created": 1605644752395, // Ignore
   "last_updated": 1605650566509// Ignore
 },
 "on_spot_update_time": 1605644298704,
 "on_spot_creation_time": 1605644298702,
 "success": true,
 "message": "Request processed successfully."
}
```

Field	Explanation
cv_file.id	This ID is a Unique SERVER Key that has been GENERATED by SERVER and sent here in response. In CV File Upload API you will have to use this ID to SUBMIT your actual CV File
cv_file.tsync_id	Returns UUID type unique indeintifier string ID for this entity – which was actually submitted in API payload.

fieldbuzz

3.3. **File upload API:**

This API allows to submit ACTUAL CV File to Field Buzz server using the FILE TOKEN ID that was received using previous API.

3.3.1. API Detail:

3.3.1.1. API URL: https://recruitment.fisdev.com/api/file-object/{FILE_TOKEN_ID}/

3.3.1.2. Request Type: POST

3.3.1.3. API Payload format: A BINARY File needs to be attached. Check below example:

Field	**Explanation**	**Restrictions to handle**
file	Form-Data, FILE Type, Multi Part	Only PDF is allowed and maximum file size 4MB.

3.3.1.4. **Successful API Response Sample**:

```
{
  "id": 1,
  "tsync_id": "be96f0e188a6de73b186c043-7365-4605-85eb-76021f16803f",
  "code": "FO-00001",
  "name": "File_OFOSDYVU_18112020_022552.",
  "path": "static_media/uploads/1/2020/11/18/File_OFOSDYVU_18112020_022552.",
  "extension": null,
  "description": null,
  "file": "http://127.0.0.1:7013/api/file-
object/1/static_media/uploads/1/2020/11/18/File_OFOSDYVU_18112020_022552.",
  "date_created": 1605644752395,
  "last_updated": 1605645158668,
  "success": true,
  "message": "Request processed successfully."
}
```

That’s All for API Documentation Section.

4. Assignment Tasks:

4.1. If you are an Backend Applicants:

4.1.1. Develop a python-based DJANGO application that can do the following:

4.1.1.1. A simple screen to take INPUT from users "all the fields" as per **SECTION 3.2.1.5**. List of fields Mentioning here again for your convenience:

name	
email	
phone	
full_address	
name_of_university	
graduation_year	
cgpa	
experience_in_months	
current_work_place_name	
applying_in	
expected_salary	
field_buzz_reference	
github_project_url	
cv_file	PDF File Input

4.1.1.2. Must **follow** the mentioned restriction in User Input Interface as mentioned in 3rd column of table **SECTION 3.2.1.5**

4.1.1.3. A Submit button below the inputs that will validate the input restrictions. Show proper ERROR message to user if any validation is failed. If all INPUT validations are successfully passed, submit the data to server as per **API 3.2** and then submit the actual BINARY FILE using **API 3.3**

4.1.1.4. We don't expect any INTERFACE for AUTHENTICATION TOKEN part as per **API 3.1.** You can internally call the AUTHENTICATION API, collect the TOKEN and then use the TOKEN while submitting the data for API 3.1 and 3.2.

4.1.1.5. BUT if you implement a working version of simple LOGIN (username, password inputs and login button) interface for AUTHENTICATION that will be appreciated.

4.2. <u>If you are an Android Applicants:</u>

4.2.1. Develop an Android app using preferably JAVA (Kotlin is acceptable) that can do the following:

4.2.1.1. A simple screen to take INPUT from users "all the fields" as per **SECTION 3.2.1.5**. List of fields Mentioning here again for your convenience:

name	
email	
phone	
full_address	
name_of_university	
graduation_year	
cgpa	
experience_in_months	
current_work_place_name	
applying_in	
expected_salary	
field_buzz_reference	
github_project_url	
cv_file	PDF File Input

4.2.1.2. Must **follow** the mentioned restriction in User Input Interface as mentioned in 3rd column of table **SECTION 3.2.1.5**

4.2.1.3. A Submit button below the inputs that will validate the input restrictions. Show proper ERROR message to user if any validation is failed. If all INPUT validations are successfully passed, submit the data to server as per **API 3.2** and then submit the actual BINARY FILE using **API 3.3**

4.2.1.4. We don't expect any INTERFACE for AUTHENTICATION TOKEN part as per **API 3.1.** You can internally call the AUTHENTICATION API, collect the TOKEN and then use the TOKEN while submitting the data for API 3.1 and 3.2.

4.2.1.5. BUT if you implement a working version of simple LOGIN (username, password inputs and login button) interface for AUTHENTICATION that will be appreciated.

5. Submission and Evolution

5.1. Submission: (same rule are applied to both backend and android applicants)

5.1.1. As per SECTION 3.2 API, from your FINAL version of **developed application**, YOU WILL HAVE TO SUBMIT YOURSELF with all relevant and true information for once. The record will be safely stored in our system's database.

5.1.2. Once you find that your application got successful response after submitting data, you can be sure enough that we will start looking into your information from our database right after the deadline of submission.

5.1.3. Any submission created after deadline won't be accepted.

5.2. Evaluation:

5.2.1. We will first check our database for your submitted information. Only valid information will be passed for further checks. And then we will,

5.2.1.1. Check and validate your uploaded CV file from our server.

5.2.1.2. And following your submitted GitHub Project URL, we will check and review your coding to evaluate your engineering skills. So, your application must be available in GITHUB with public access, so our team can access your GitHub project code.

5.2.2. And last but not the least, **"People who copy are always one step behind and get exposed one step ahead"**