4b Technical report - API

Development of the new Worker mobile app has led to a new Rest API Interface introduced into the CEDAH. The Rest calls are specified below. Additional POST/PUT data has not been included (apart from login) but we can provide the information if required.

Login/Authenticate

Method	HTTP method	Endpoint	Description
get	POST	/api/restlogin	Login

Worker Details

Method	HTTP method	Endpoint	Description
get	GET	/api/user/userId	Gets a worker's basic details

Worker Availability

Method	HTTP method	Endpoint	Description
list	GET	/api/user/***userId***/availability	Gets a worker's availability
add	POST	/api/user/***userId***/availability/***availabilityId***	Add availability
delete	DELETE	/api/user/***userId***/availability/***availabilityId***	Delete availability
update	POST	/api/user/***userId***/availability/***availabilityId***	Update availability

Worker Jobs

Method	HTTP method	Endpoint	Description
list	GET	/api/user/***userId***/jobs	Gets a worker's jobs
accept	PUT	/api/user/***userId***/job/***jobId***	Accept a job
reject	DELETE	/api/user/***userId***/job/***jobId***	Reject a job

Worker Roles

Method	HTTP method	Endpoint	Description
list	GET	/api/user/***userId***/roles	Gets a worker's roles
accept	PUT	/api/user/***userId***/role/***roleId***	Accept a role
decline	DELETE	/api/user/***userId***/role/***roleId***	Reject a role

Worker Timesheets

Method	HTTP method	Endpoint	Description
list	GET	/api/user/***userId***/timesheets	Gets a worker's timesheets
accept	PUT	/api/user/***userId***/timesheet/***timesheetId***	Accept a timesheet
reject	DELETE	/api/user/***userId***/timesheet/***timesheetId***	Reject a timesheet

An Example of the json data sent with the login would be:

```
{
    "username" : "myEmail@somewhere.com",
    "password" : "SomeOldPassword"
}
```

This would (if successful) return a status of “success” along with a token to be used in successive calls to the system.
Jar updates/Gradle build
Version control of all source code uses Git hosted by bitbucket (owned by Atlassian). This allows (amongst others):

- automated software builds (Gradle)
- simple deployment to multiple systems
- automated testing
- piplelines for separate software development & testing

The gradle build provides an easier way to reference jar files used in the build. They are specified in a build file, rather than including each file in our application source. It is simpler to just specify the version or each jar file (in the build file) rather than downloading every new jar file when a new version is produced.

We are currently at a “half-way house”, with some jars included and others not. Ideally we would like to move all jar files to the gradle build file. This would also include the test file jars.
Some jar files currently used by the CEDAH are old. In particular Spring files which provides our comprehensive programming and configuration model is still on version 4. The current version is 5.1.8.

Due to changes to methods and API calls, it needs additional time to update.
This is also the issue with other areas in the code including struts 1 and Bootstrap 2.