

Enterprise Computing Coursework (2014/2015)

Professor Stephen Gilmore
School of Informatics
The University of Edinburgh

January 15, 2015

1 Coursework

- The goal of the coursework is to make use of a newly-released Open Data API made available by Transport for Edinburgh.
- The API is available at <http://tfe-opendata.readme.io/v1.0>
- From the API you can request data about stop locations, service routes, journey planning, timetables and live bus locations for the city of Edinburgh.
- To access the API and use it you need an API key which looks like this:0c627af5849e23b003073525508¹

1.1 Requirements

1.1.1 Functional requirements

- The functional requirements for the project are intentionally open. You are required to use the data which is made available by the Transport for Edinburgh Open Data web service, but *it is not specified how you are to use it*.
- Underspecification such as this is very common in practical software development. In contrast, it is very rare indeed to receive a fully formal specification which details all of the development work which is to be done.
- The guidance that you have been given is that *you should make the open data information accessible*. The data gives you the potential to do something: you decide what.

¹Some of the letters and numbers of the API key have been blanked out. If you were at the lecture then you heard me say which letters and numbers these were.

- As is often the case in practical software development, for the software development part of this course you have some freedoms which you should use as you see fit, and some constraints which you just have to learn to live with.
- You are free to choose the technology which you use to implement your system on the server side (if you even have a server side). You could use Python, PHP, Java, or another language of your choice.
- You are constrained to use TypeScript on the client side. TypeScript is a typed dialect of the JavaScript language.

1.1.2 Non-functional requirements

Ten non-functional requirements will be revealed as the course lectures progress. These non-functional requirements will be delivered verbally, and not written down on slides. It is your job to write them down in your own words.

When you submit your coursework for assessment you will be required to say what these ten non-functional requirements are, and whether or not you think that your submission meets these non-functional requirements.

Non-functional requirement #1 was announced in lecture 1.

2 Teams

Coursework is undertaken in a team. There are twelve coursework teams in the 2014/2105 session. Teams may have at most five members, or maybe six.

Teams were decided in advance by the course lecturer, and are non-negotiable. Each team has a team leader who has a casting vote for decision-making purposes.

Teams should organise for themselves a way of collaborative working which suits them best (in terms of software repositories, communication, mailing lists, and other administrative matters).

Although students on the course receive an individual mark for their work, it is in everyone's best interest to ensure that the submission of the team as a whole is successful.

3 Timetable of events

The coursework will proceed according to the following timetable.

Thursday 15th January 2015	Coursework released
Thursday 12th February 2015 at 16:00	Part 1 due
Thursday 5th March 2015	Part 1 feedback received
Thursday 2nd April 2015 at 16:00	Part 2 due
Thursday 23rd April 2015	Part 2 assessment received

4 For Part 1

Part 1 is zero-weighted: it is just for feedback. For Part 1 of the coursework you should deliver a draft or design of your proposed software system. You should describe the functionality of the system and record the technologies which will be used to implement it. You should document which members of your team plan to contribute to which parts of the software.

You should expect to receive feedback on your Part 1 submission no later than 15 working days after the deadline for submission for Part 1. This feedback will take the form of an email with an attached document, sent to all members of a group. The same message is received by all members of the group. Feedback from Part 1 makes suggestions for improvement to your submission, feeding into the work which you submit for assessment in Part 2.

5 For Part 2

Part 2 accounts for 100% of the coursework mark. For Part 2 of the coursework you should deliver your completed software system together with documentation.

The team leader in each team is responsible for making the submission. Team members are encouraged to ensure that the team leader does this. Early submission is no problem at all. Multiple submissions are also allowed: later submissions will overwrite earlier ones. The accompanying documentation for the submission should make clear the contribution of each team member.

You will receive an individual email message giving you written feedback on your Part 2 submission. Feedback on Part 2 is *summative*. It gives an explanation of how the grade for your practical was derived in general high-level terms (“you lost marks here” and “you gained marks here”). In addition there will be formative feedback at this stage which is hoped will be useful for future courses or future enterprises which involve some element of software development.