

# Datenbanken und Web-Techniken

## Project Task

### Summer Semester 2019

# 1. Introduction

Many websites offer their content in addition to the actual HTML pages as web feeds. The complete content can be embedded in the feed, but mostly only excerpts are offered and link to the actual HTML pages. Often these web feeds also do not contain the information on all available content, but merely link to a certain number of the latest posts. Users can subscribe to and aggregate these web feeds from multiple providers to access all content through a common interface.

The aim of this practical task is to develop a web application that aggregates various web feeds and presents them in a common interface. The exact task description follows on the next pages.

## 2. Task Description

### 2.1 Preliminary remarks

The following practical task may be processed in groups of up to two participants. If more than one participant is present, it should be indicated for all parts (programming, term paper), who was responsible for which part. For groups, the additional tasks mentioned at the end are obligatory (individuals may of course solve them, too).

The submission consists of:

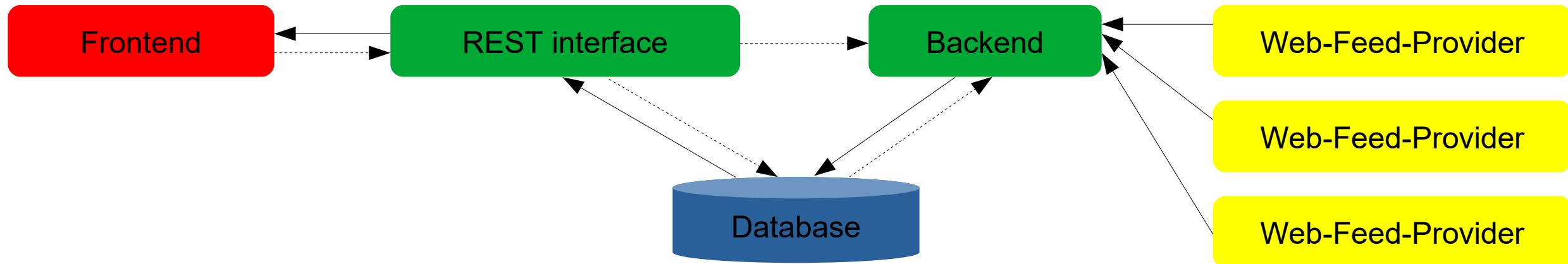
- 1.a PDF-file of the term paper in paper format A4 or A5 and
- 2.a separate ZIP-archive that contains the sources and a small manual on how to use them. The archive may not exceed a size of 10MiB.

## 2. Task Description

### 2.2 Programming

The program consists of

- 1.a backend for aggregating and processing the data of the web feeds,
- 2.a database for storing the data of the web feeds,
- 3.a REST based interface for providing the data from the database and
- 4.a frontend for displaying the data from the database.



## 2. Task Description

### 2.2.1 Backend

- 1.The backend aggregates and processes the data of the various web feeds.
- 2.It can handle multiple standards of web feeds, supporting at least one version of RSS and the Atom format.
- 3.It prevents the updating of web feeds in a too small time interval, so that it waits at least 10 minutes between two requests for a web feed (and blocks other requests to do so).
- 4.It ensures a consistent storage of the data in a database.
- 5.Existing records can also be updated, but no duplicates are created.
- 6.Records that have reached a certain age are deleted and no longer inserted. By default, this age should be 30 days.

## 2. Task Description

### 2.2.2 Database

- 1.The database stores the aggregated and processed data of the web feeds.
- 2.It can also be used to store the configuration.
- 3.The frontend must never communicate directly with the database.

## 2. Task Description

### 2.2.3 REST based interface

- 1.The REST based interface provides communication between the frontend and the database.
- 2.It delivers the data from the database to the frontend.
- 3.It can also be used to store configuration settings in the database.
- 4.Furthermore, it is possible to use the REST based interface to initiate an update of the web feeds by the backend. Therefore, it is not necessary to implement a proactive service which automatically performs the update.

## 2. Task Description

### 2.2.4 Frontend

1. The information of all web feeds can be displayed. The following details must at least be available:
  - a) title of the content
  - b) clickable link to the webpage with the actual content
  - c) date when the content was published according to the web feed
  - d) date when the entry in the web feed first got detected by the program
  - e) provider where the content is available
2. The display of the contents of individual providers, i.e. web feed links, can be enabled or disabled.
3. Usability aspects are taken into account.
4. The design is appealing.



## 2. Task Description

### 2.2.4 Frontend

5.The following information can be displayed for each provider:

- a)address (URL) of the web feed
- b)date of the last successful update of the web feed
- c)Date of the latest record according to information from the web feed (the date should also be available if all records are older than the threshold for deleting old records and therefore no record of the web feed is in the database anymore)
- d)date of the last attempt to update with the number of records found or an indication of an error that occurred during the update attempt

6.At least five different working web feeds and a malfunctioning web feed are available for demonstration purposes.

## 2. Task Description

### 2.2.5 Additional task for groups

1. The backend can also handle "broken" web feeds that do not fully adhere to the labelled standard.
2. For each record, there is a detail view that shows the contents included in the web feed.
3. The time interval for the web feed update blockade and the age for deleting old records can be set via the frontend.
4. The complete web feed management takes place via the frontend, whereby at least the following options exist beyond the previous display options:
  1. Insert new web feeds with address (URL) and display name, which is also used in the overview
  2. Modify the entry of a web feed (change the URL or display name) while preserving existing records in the database
  3. Delete the entry of a web feed, deleting all associated data from the database
5. It is possible to export the data of the web feeds selected in the frontend view as a new web feed in RSS or Atom format. This export should also be reachable via a normal URL with a GET request and contain all stored data. Ultimately, this should make it possible to transform a web feed from RSS to Atom format or vice versa.

## 2. Task Description

### 2.3 Term Paper

A term paper is to be written, that satisfies the following conditions:

- 1.Amount of about 10 pages (content!) for each project participant.
- 2.On the cover page there is the name, study course and matriculation number of all students recorded.
- 3.An overview of the utilized technologies is given and why they were used. They will be also classified in the context of the lecture.
- 4.The project is presented.
- 5.A good form is kept and a balanced ratio of pictures and text is payed attention to.
- 6.In the Appendix (i.e. not part of the 10 pages) an API documentation is inserted, which contains the following:
  - a)list of the functions of the REST based interface
  - b)for every function the list of the parameters and the return values each with type and meaning
  - c)for complex structures also the inner structure is to be documented respectively
  - d)Documenting by only listing of examples is not sufficient!

## 3. Examination

The examination consists of a 10-minute presentation, which should meet the same criteria as the content of the term paper, but the API documentation is omitted here.

So the project and the used technologies should be presented and put into the lecture context. The presentation should also include a live demonstration of the project or a demonstration video demonstrating all parts of the practical task. For group work, the presentation time should be divided equally between both students.

After that some questions – concerning the project, but also out of the lecture – are given. Finally there will be a short consultation and you will be informed about your mark.

## 4. Dates

- Handout of task description:
  - starting 2019-05-20 00:00 MEST (UTC+2)
- Submission of project:
  - until 2019-06-23 23:59:59,9 MEST (UTC+2)
  - per E-Mail
    - to [Daniel.Richter@informatik.tu-chemnitz.de](mailto:Daniel.Richter@informatik.tu-chemnitz.de)
    - by mentioning all participants
    - including PDF-documentation and ZIP-archive as attachment
- Oral exam and talk:
  - between 1. and 12. July
  - Modalities of appointment allocation will be announced after the project has been submitted