

Промежуточный отчёт по проекту Shared IDE

Исхаков Тимур

27 марта 2016 г.

1 Текущая архитектура

Для синхронизированной работы вместо GET и POST запросов используется протокол WebSocket.

Django — синхронный фреймворк, поэтому асинхронный WebSocket нельзя использовать с Django нативно. Для осуществления работы WebSocket используется Redis сервер, который подклчается в Django, подменяя её wsgi-приложение.

Для создания асинхронных задач из Django (обработчика сообщений на стороне сервера, который будет получать изменения клиентов через WebSocket) используется Celery.

В настоящий момент изменения отправляются broadcast-сообщением всем, открывшим данный документ (`/ide/<document>`), и применяются отдельно клиентом и сервером.

Сторона клиента реализована с помощью фреймворка React, библиотеки Operational Transformation.

2 Сборка JS, JSX

В проекте используются JS и JSX по стандарту ECMAScript 6 на стороне клиента. В файлах имеются зависимости: `«import»`, `«export»`.

К сожалению, ни JS, ни зависимости не поддерживаются современными браузерами, поэтому файлы, используемые на стороне клиента, собираются в момент сборки в монолитные JS-файлы:

1. babel: JSX → JS, раскрытие ECMAScript 6
2. browserify: разрешение зависимостей

Собранные файлы из `./src/` оказываются в `./static/`.

Конфигурация сборки находится в файле `./babelrc`, необходимые пакеты — в файле `./package.json`.

Сборка занимает порядка минуты, поэтому она кешируется: хеши собранных файлов хранятся в файле `./js_build_hash.json`.

3 Дальнейшая работа

1. Хранилище файлов, его интерфейс;
2. Сохранение изменений на сервере;
3. Пользовательские политики при изменении файлов:
 - (a) Разрешение публикации в WebSocket только редактором файла (если это возможно в Redis, иначе вся передача данных будет по схеме редактор → сервер → читатели;
 - (b) Подтверждение личности редактора.
4. Запуск кода на тестах.

4 Установка

1. Установка пакетов:

```
pip3 install -r requirements.txt
npm install
```
2. Настойка Django, сборка базы данных:

```
./manage.py makemigrations
./manage.py migrate
./manage.py createsuperuser
```

5 Запуск

1. Запуск Redis сервера:

```
redis-server
```
2. Запуск Celery сервера:

```
./manage.py celery -A shared_idc worker
```
3. Запуск Django сервера:

```
./manage.py runserver
```