

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Ярославский государственный университет им.
П.Г. Демидова»

Кафедра информационных и сетевых технологий

Сдано на кафедру

«_____» _____ 2020 г.

Заведующий кафедрой,
к. ф.-м. н.

_____ Д. Ю. Чалый

Выпускная квалификационная работа

**Разработка клиентской части системы для
автоматизации процесса рекрутирования
сотрудников**

по направлению
09.03.03 Прикладная информатика

Научный руководитель
стар. преподаватель

_____ Н. В. Легков

«_____» _____ 2020 г.

Студент группы ПИЭ-41БО

_____ О. С. Гаршина

«_____» _____ 2020 г.

Ярославль, 2020

Реферат

Объем 11 с., 4 гл., 0 рис., 0 табл., 1 источников, 0 прил.

Ключевые слова: **react, рекрутер, автоматизация, HR, резюме, front-end, JavaScript**

Текст реферата должен отражать объект исследования, цель работы, результаты работы, область применения, степень внедрения или рекомендации по внедрению.

Содержание

Введение	4
1. Теория	5
2. О задаче	6
2.1. Постановка задачи	6
2.2. Требуемый функционал	6
2.3. Используемые программные средства	6
3. Решение задачи	7
3.1. Подготовка окружения расширения	7
3.2. Подсветка синтаксиса и авто дополнение по сокращениям	8
3.3. Компиляция умных контрактов	8
3.4. Развертывание умных контрактов в локальной сети	8
3.5. Развертывание умных контрактов в удаленных сетях	8
3.6. Отладка умных контрактов	8
3.7. Визуальный интерфейс для взаимодействия с развернутыми умными контрактами	8
4. Результаты решения задачи	9
Заключение	10
Список литературы	11

Введение

Во введении обосновывается актуальность выбранной темы, описываются объект и предмет исследования, цели и задачи, методы исследования и приводится краткое описание структуры работы.

1. Теория

2. О задаче

2.1. Постановка задачи

Требуется создать расширение для интегрированной среды разработки Visual Studio Code, позволяющее создавать, собирать, отлаживать, разворачивать и взаимодействовать с умными контрактами в сети Ethereum

2.2. Требуемый функционал

1. Подсветка синтаксиса умных контрактов и автоматическое дополнение ключевых слов по сокращениям
2. Компиляция умных контрактов
3. Развертывание умных контрактов в локальной сети
4. Развертывание умных контрактов в удаленных сетях
5. Отладка умных контрактов
6. Визуальный интерфейс для взаимодействия с развернутыми умными контрактами

2.3. Используемые программные средства

Основываясь на том, что расширение пишется для IDE VSCode, для разработки были выбраны следующие программные средства:

- VSCode для разработки и отладки расширения
- TypeScript в качестве основного языка программирования
- JavaScript в качестве языка программирования при создании графического интерфейса взаимодействия в методами умного контракта
- Truffle для компиляции, отправки кода контракта в цепочку блоков
- Ganache для создания локальной сети Ethereum
- Drizzle в качестве прослойки между актуальным состоянием развернутого контракта в сети и его отображением в внутреннем состоянии JavaScript
- React.js для создания визуального интерфейса взаимодействия с умными контрактами

3. Решение задачи

3.1. Подготовка окружения расширения

Компания Microsoft советует использовать утилиту `yo` в связке с `generator-code`. `yo` отвечает за запуск пошагового руководства для первоначальной настройки структуры, а `generator-code` предоставляет сами шаги настройки. Этим инструментарием было решено и воспользоваться.

После выполнения команды `yo code` и прохождения по предлагаемым шагам настройки получаем следующую структуру проекта:

```
/
├── .vscode
│   ├── launch.json
│   └── tasks.json
├── .gitignore
├── README.md
├── src
│   └── extension.ts
├── package.json
└── tsconfig.json
```

- `launch.json` отвечает за конфигурацию запуска разрабатываемого расширения в режиме отладки
- `tasks.json` отвечает за описание задач, вызываемых из `launch.json`
- `extension.ts` является точкой запуска расширения, в нем принято регистрировать обработчики события, вызываемых пользователем расширения
- `package.json`
- `tsconfig.json`

- 3.2. Подсветка синтаксиса и авто дополнение по сокращениям**
- 3.3. Компиляция умных контрактов**
- 3.4. Развертывание умных контрактов в локальной сети**
- 3.5. Развертывание умных контрактов в удаленных сетях**
- 3.6. Отладка умных контрактов**
- 3.7. Визуальный интерфейс для взаимодействия с развернутыми умными контрактами**

4. Результаты решения задачи

В результате решения задачи было получено расширение для VSCode

Заключение

Список литературы

- [1] Visual Studio Code - Code Editing. Redefined URL: <https://code.visualstudio.com>
(дата доступа: 09.06.2020)