

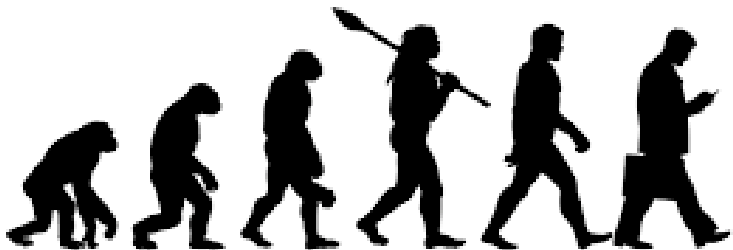
Министерство науки и образования РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Заполярный государственный университет им. Н. М. Федоровского»
Факультет электроэнергетики, экономики и управления
Кафедра информационных систем и технологий

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Тема: "Инженерный старт проекта"

Зачем нам “инженерный старт”?

- В проекте важен не только “код работает”, но и как его менять без боли
- Уходим от проблем “а у меня работает”
- Прикладываем часть рутины на инструменты



Стандарт оформление и ведения репозитория

- API и клиент отдельно
- Обязательно наличие README.md, .gitignore
- pre-commit
- Использование форматеров и линтеров
- Зафиксированная культура комитов

Файловая структура API и REST паттерн CSR

Контроллер-Сервис-Репозиторий

Контроллер (Controller)
Управление REST-интерфейсом и
вызов бизнес-логики

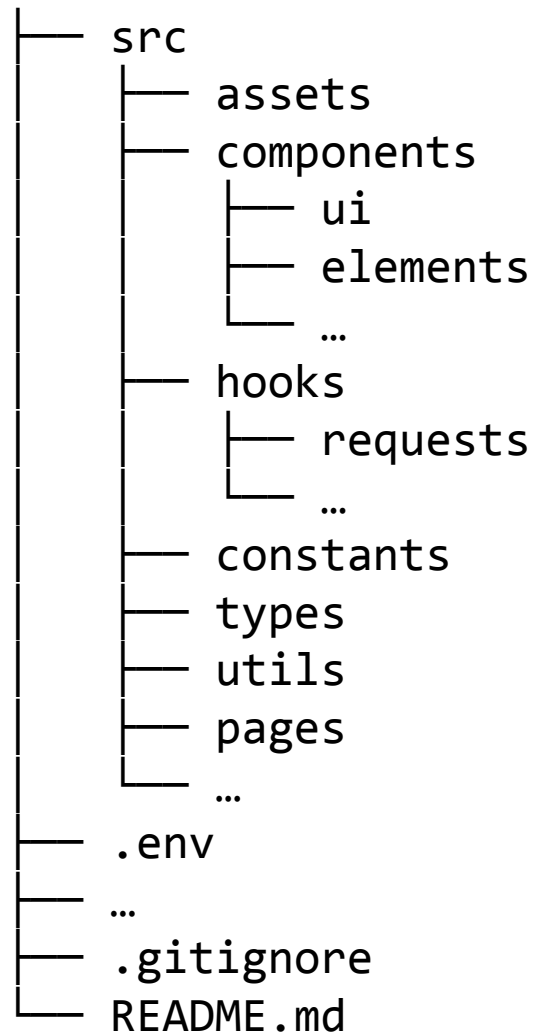
Сервис (Service)
Реализация бизнес-логики

Репозиторий (Repository)
Хранение и доступ к данным
(сущностям) в системе

Файловая структура

```
├── app
│   ├── user
│   │   ├── controllers
│   │   ├── ...
│   │   ├── services
│   │   ├── ...
│   │   └── repositories
│   └── ...
├── .env
├── ...
├── .gitignore
└── README.md
```

Файловая структура фронтенд-приложения



Форматеры и линтеры

- Formatter: делает код одинаковым (скобки, переносы, пробелы)
- Linter: ловит проблемы и плохие практики (unused, подозрительные места)

Конкретные инструменты

- Python: ruff (линтер + форматтер), для смелых mypy (типизация)
- JS/TS: prettier (форматтер), eslint (линтер)

Стандарты

- Python: PEP 8 (стиль кода), PEP 257 (docstring'и)
- js/ts:
- Go: вшитый
- C/C++: пу пу пу

Pre-commit

Pre-commit – это набор команд которые выполняются перед каждым комитом.

С его помощью можно поддерживать чистоту репозитория (писать в своем стиле а комитить в общепринятом, очищать не используемы код, проверять файлы на валидность) и поддерживать работоспособность проекта (авто тесты)

Культура коммитов в Git-репозитории

- Коммиты должны быть атомарны
- Примерная структура – суть действия + где сделал + описание что именно сделал (при необходимости)
- Для фичи – ветка. Для конкретных реализаций в рамках фичи – комит

Ведение документации

- Инструкция по развертыванию и запуску
- Описание структуры проекта
- Описание используемых контрактов
- Описание используемых технологий и “подводных камней”

Требования по дисциплине

Допуск к экзамену:

- Соответствие итогового проекта по стандартам

Начальные требования к итоговому проекту

- Реализация REST API
- Наличие настроенных линтеров и форматоров (на клиенте и API)
- README.md с подробным описанием структуры проекта, настройка и тп
- Оформление абсолютно всех файлов проекта должно быть стандартизировано
- Наличие настроенного pre-commit. Комит не должен отправиться если есть большие файлы, есть маркеры не решенных конфликтов, содержимое файла не соответствует стандарту, в коде присутствуют не использованные импорты, переменные или мусорный код (допускается чтобы скрипт сам "подчищал код" но не отправлял комит без проверки разработчиком), названия файлов не соответствует стандарту

Министерство науки и образования РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Заполярный государственный университет им. Н. М. Федоровского»
Факультет электроэнергетики, экономики и управления
Кафедра информационных систем и технологий

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Тема: "Инженерный старт проекта"