1. Хостинг веб-приложения на сервисе Render

Целью данной лабораторной работы является включение имеющейся клиентской части (лабораторных работ прошлого семестра) внутрь нового приложения, которое будет развернуто в сервисе облачного хостинга Render.

Контрольный срок сдачи работы: 4-ая неделя обучения.

Для выполнения данной лабораторной работы вам необходимо:

1) Установить NodeJS (LTS-версия);
2) Установить пакет NestJS CLI глобально через пакетный менеджер;
3) Зарегистрироваться в хостинг-сервисе Render;

Чтобы создать приложение NestJS с помощью Nest CLI, необходимо выполнить команду `nest new` в терминале вашей операционной системы. Укажите имя проекта и выберите пакетный менеджер, которым вы пользуетесь (по умолчанию npm). В результате будет сгенерирован проект в новой директории с именем вашего проекта.

Откройте проект в IDE и ознакомьтесь с файлом `package.json`. Отредактируйте поле `author` согласно схеме, укажите ваше авторство.

Добавьте явное указание используемой версии NodeJS согласно схеме, без этого сборка вашего приложения на сервисе Render упадет с ошибкой.

Проверьте, что ваше приложение работает, для этого выполните скрипт запуска, описанный в `package.json` (`npm run start` в случае, если вы пользуетесь пакетным менеджером по умолчанию).

Обратите внимание, что запущенное приложение запускается на 3000-ом порту: http://localhost:3000. Для хостинга Render такое поведение приемлемо, т.к. Render автоматически найдёт открываемый вами порт, но с общепринятой практике обычно хостинг сам сообщает вам через переменную окружения с именем `PORT` номер порта, на котором необходимо принимать соединения. Биндинг вашего приложения на 3000-ый порт осуществляется в файле `src/main.ts`. Отредактируйте файл таким образом, чтобы приложение начинало слушать подключения на порту, указанному в переменной окружения, а в случаях, когда порт не указан, брать порт по умолчанию (на ваш выбор).

Переменные окружения хоста могут быть прочитаны двумя способами:

1) Напрямую средствами NodeJS через `process.env`;
2) С помощью модуля конфигурации Nest.

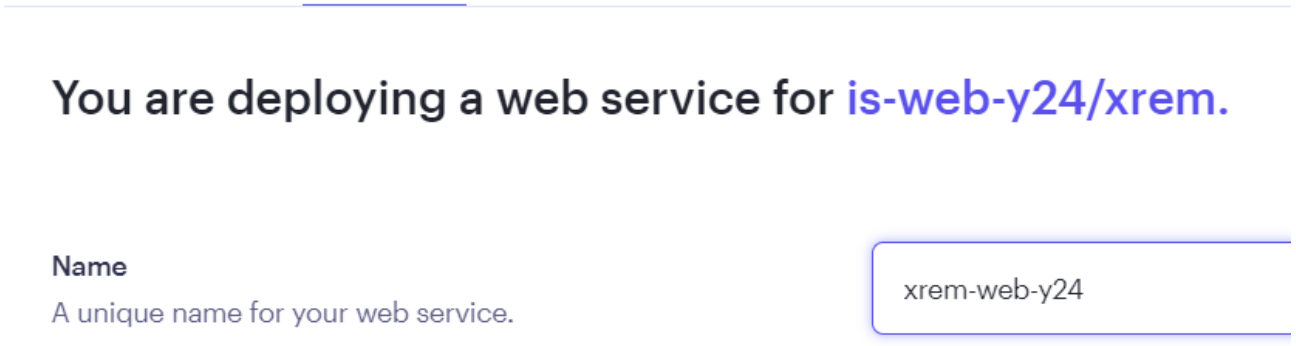
После добавления необходимой обработки проверьте, что приложение стартует на указанном в переменной окружения порту, для этого необходимо настроить профиль запуска (пример для WebStorm).

Если всё получилось, то можно переходить к следующему шагу — созданию самого инстанса приложения на хостинге. Для этого необходимо загрузить сгенерированное с помощью Nest CLI шаблонное приложение в GitHub репозиторий. Обратите внимание, что в отличии от прошлого семестра, **приватный (!)** репозиторий необходимо создать в организации is-web-y24.

Как попасть в организацию?

Для этого необходимо заполнить форму (https://is-web-y24-invite.onrender.com/) и предоставить ссылку на ваше приложение, это нужно, чтобы преподаватели могли отслеживать ваш прогресс в течение семестра. После авторизации вы сразу же получите приглашение в организацию.

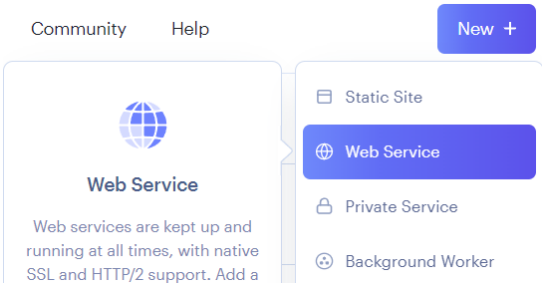
Application URL заполняется по форме = имя + '.onrender.com'



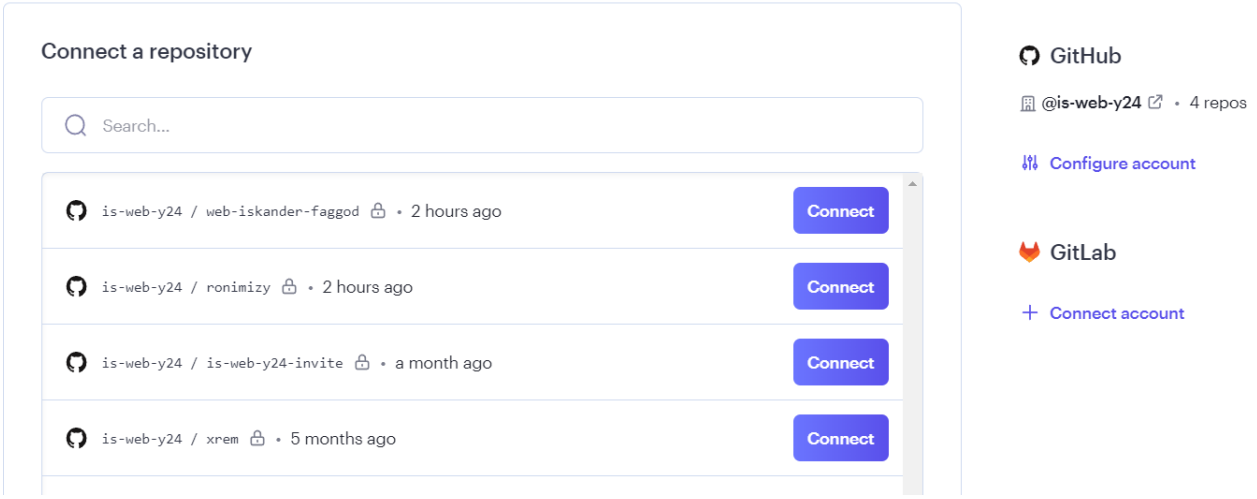
Например: https://xrem-web-y24.onrender.com/

Название приложения будет использовано в качестве поддомена.

После того как вы загрузите код приложения в репозиторий, необходимо его будет подключить к хостингу Render. Для этого выбирайте в меню **New + Web Service.**



В организации уже настроена интеграция с хостинг сервисом, поэтому дополнительно вам ничего подключать не нужно, просто выбирайте ваш репозиторий и переходите на следующий шаг.



Имя сервиса = То что вы указывали в форме когда добавлялись в организацию.

Регион = на ваш выбор, рекомендую ближайшую геопозицию,
Europe, Frankfurt

Ветка = В зависимости от того как привыкли, **main** либо **master** (либо **development** если планируется Git Flow и более чем одна версия среды)

Корневая директория = директория в которой находится ваш проект, если не меняли файловую структуру, то оставьте поле пустым.

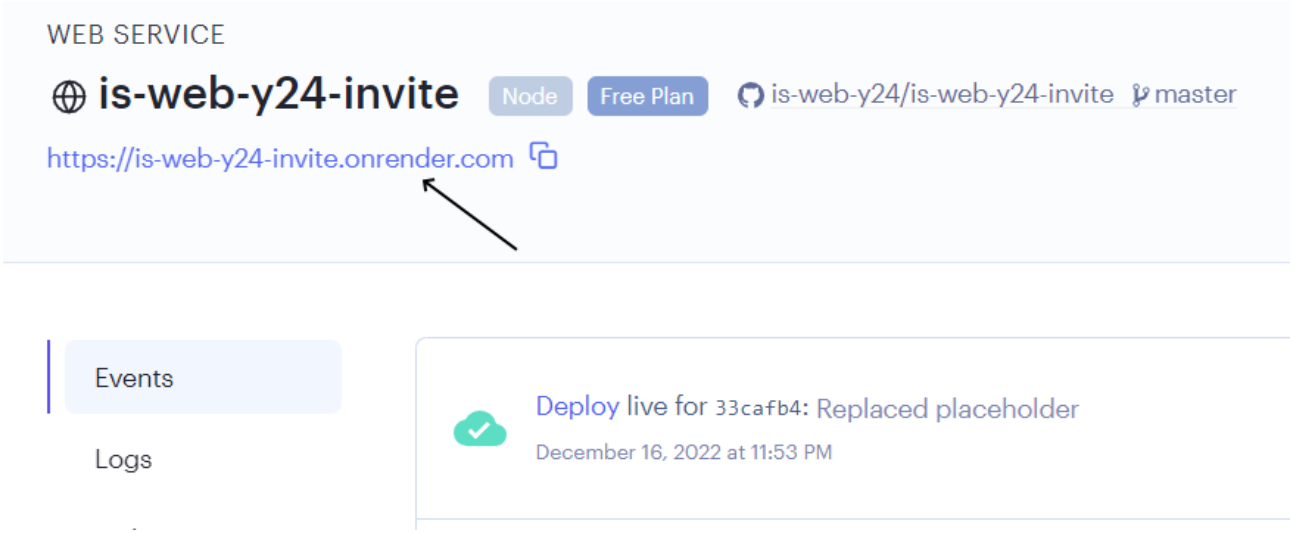
Окружение = Node

Команда для сборки = Необходимо восстановить зависимости и вызвать скрипт build с помощью выбранного вами пакетного менеджера.
Для npm: npm install; npm run build
Для yarn: yarn; yarn build

В качестве входной точки рекомендуется использовать production-версию вашего приложения, т.е. команду `npm run start:prod` для npm
либо команду `yarn start:prod` для yarn соответственно.

После того, как вы создадите веб сервис на хостинге, должен начаться автоматический процесс сборки с отображением прогресса.

Если все прошло без ошибок, то ваше приложение должно быть доступно в сети Интернет, по ссылке предоставленной на хостинге



Если при переходе на страницу вы видите Hello World, то всё отлично, осталось только разместить статические ресурсы (на данный момент статикой будет являться весь фронтенд, который был разработан в рамках лабораторных работ прошлого семестра) и научится отдавать их с сервера. Для этого необходимо воспользоваться следующим рецептом.

Укажем в момент запуска нашего Nest-приложения, что мы хотим воспользоваться шаблоном MVC, основанном на фреймворке Express. Укажем, что хотим иметь возможность отдавать статические ресурсы, расположенные в конкретной папке (по умолчанию `public`). Затем необходимо будет перенести все ваши файлы с frontend'ом в указанную папку и проверить, что статика отдаётся, т.е. сделать запрос на http://localhost:port/index.html.

Если всё окей, измените стандартный readme.md файл и добавьте краткое описание вашего проекта, укажите ваше авторство.

Далее делаем коммит и пушим результат.

Процесс доставки изменений (Continuous Delivery) на хостинг должен начаться автоматический, точно таким же образом как это происходило при использовании GitHub Pages.

После проделанных шагов можно считать лабораторную работу выполненной, остается только пройти защиту на практическом занятии.

Следующие лабораторные работы ищите в публичном репозитории внутри организации.