**Экзаменационные вопросы**

**по дисциплине «Программирование серверных кроссплатформенных приложений» для студентов 3-го курса специальности ПОИТ**

1. Протокол HTTP, основные свойства HTTP, структура запроса и ответа, методы, статус (серии значений), заголовки, параметры. Протокол HTTPS. Понятие web-приложения, структура и принципы работы web-приложения. Понятие асинхронности.
2. Платформа Node.js, версии, назначение, основные свойства, структура, принципы работы, основные встроенные модули и их назначение, применение внешних модулей (пакетов).
3. HTTP-аутентификация (Basic, Digest, Forms).
4. Протокол HTTPS. Протокол TSL. Сертификаты. Взаимодействие центра сертификации и владельца защищенного ресурса.
5. Протокол WebSocket, основные свойства, процедура установки соединения. WebSockets API.
6. Применение СУБД Redis. Основные принципы работы.
7. Применение пакета Sequelize. Основные принципы работы (подключение, объявление моделей, конфигурация, подход Code First, поисковые методы, фильтрация, пагинация, сортировка, группировка). Пример.
8. Применение пакета Sequelize. Основные принципы работы (добавление, изменение, удаление строк в таблице, raw query). Пример.
9. Применение пакета Sequelize. Основные принципы работы (scopes, хуки, соединения, отношения между таблицами, транзакции). Пример.
10. Пакет Express. Основные принципы работы. Маршрутизация, промежуточные обработчики, обработка ошибок. Пример.
11. Пакет Express. Основные принципы работы. Статические файлы. Отдача статики. Кэширование на стороне клиента.
12. Пакет Express. Основные принципы работы. Объекты запроса и ответа.
13. Пакет Express. Основные принципы работы. Обработка query- и path-параметров GET-запроса. Пример.
14. Пакет Express. Основные принципы работы. Обработка тела (form, json) POST-запроса. Пример.
15. Пакет Express. Основные принципы работы (методы download и attachment, переадресация). Пример.
16. Handlebars. Назначение, использование: шаблоны, макеты, хэлперы, частичные представления.
17. Пакет Express. Основные принципы работы. Обработка Cookie. Signed cookie. Пример.
18. Пакет Express. Основные принципы работы. Применение объекта Session. Пример.
19. Аутентификация. Разработка приложения с http-аутентификацией. Пример.
20. Аутентификация. Разработка приложения с digest-аутентификацией. Пример.
21. Аутентификация. Разработка приложения с forms-аутентификацией. Пример.
22. Токен-аутентификация.
23. OAuth 2.0, OpenID.
24. Авторизация: виды, реализация.
25. Библиотеки Node ACL и CASL.
26. Сервис Heroku: устройство, ресурсы, функции. Непрерывная интеграция и доставка (CI/CD).
27. Пакет crypto. Основные принципы работы (шифрование, поточное шифрование, алгоритм Диффи-Хеллмана).
28. Пакет crypto. Основные принципы работы (хэширование, поточное хэширование, hmac).
29. Пакет crypto. Основные принципы работы (цифровая подпись).
30. Протокол WebDav. Назначение, применение, стандарты, методы. Разработка приложения с применением WebDav.
31. Протокол JSON-RPC. Назначение, структура запроса, структура ответа. Разработка клиент-серверного приложения, использующего протокол JSON-RPC. Пример.
32. WASM. Назначение, принципы использования, emcc. Разработка клиент-серверного приложения с применением WASM на стороне браузера. Пример(WasmFiddle-компиляция).
33. WASM. Назначение, принципы использования, emcc. Разработка клиент-серверного приложения с применением WASM на стороне сервера Node.js. Пример(WasmFiddle-компиляция).
34. Telegram Bot. Long pooling, webhook, принцип работы.

**Перечень лабораторных работ**

1. Лабораторная 17 (REDIS).
2. Лабораторная 18 (SEQUELIZE).
3. Лабораторная 19 (EXPRESS\_MVC)
4. Лабораторная 20 (HBS).
5. Лабораторная 21 (Basic, Digest).
6. Лабораторная 22 (Forms).
7. Лабораторная 23 (OAuth).
8. Лабораторная 24 (CASL).
9. Лабораторная 25 (HTTPS).
10. Лабораторная 26 (CRYPTO).
11. Лабораторная 27 (WEBDAV).
12. Лабораторная 28 (JSONRPC).
13. Лабораторная 29 (WASM).
14. Лабораторная 30 (TLGBOT).
15. Лабораторная 31 (SWAGGER).