## **Условие 1**

Создать веб-сервис возвращающий список задач.

URL веб-сервиса: /api/v1/task

Информация о каждой задаче должна содержать поля:

| id | Идентификатор задачи |
| --- | --- |
| title | Заголовок задачи |
| datе | Дата выполнения задачи |

Количество возвращаемых задач: 1000.

Сам массив задач для простоты можно генерить прямо в скрипте. Пусть это будет:

| id | title | datе |
| --- | --- | --- |
| 1 | Задача 1 | Текущая дата + 1 час |
| 2 | Задача 2 | Текущая дата + 2 часа |
| …. | …. | …. |
| 1000 | Задача 1000 | Текущая дата + 1000 часов |

**Дополнительно**: предположим, что генерация списка задач - трудоемкая операция, а обновляются они не чаще одного раза в час.

**Условие 2**

Создать веб-сервис, возвращающий детальную информацию о задаче.

URL веб-сервиса: /api/v1/task/:id, где id - идентификатор задачи.

Ответ должен содержать следующие поля:

| id | Идентификатор задачи |
| --- | --- |
| title | Заголовок задачи |
| date | Дата выполнения |
| author | Автор задачи |
| status | Статус задачи |
| description | Описание задачи |

Генерация информации о задаче - аналогично предыдущему заданию. Например, информация о задаче с id 87 должна содержать:

| id | title | date | author | status | description |
| --- | --- | --- | --- | --- | --- |
| 87 | Задача 87 | Текущая дата + 87 часов | Автор 87 | Статус 87 | Описание 87 |

**Условие 3**

Создать страницу отображения списка задач.

URL страницы: /

При заходе на страницу должно происходить обращение к веб-сервису получения списка задач и формироваться сам список. Это может выглядеть примерно так:

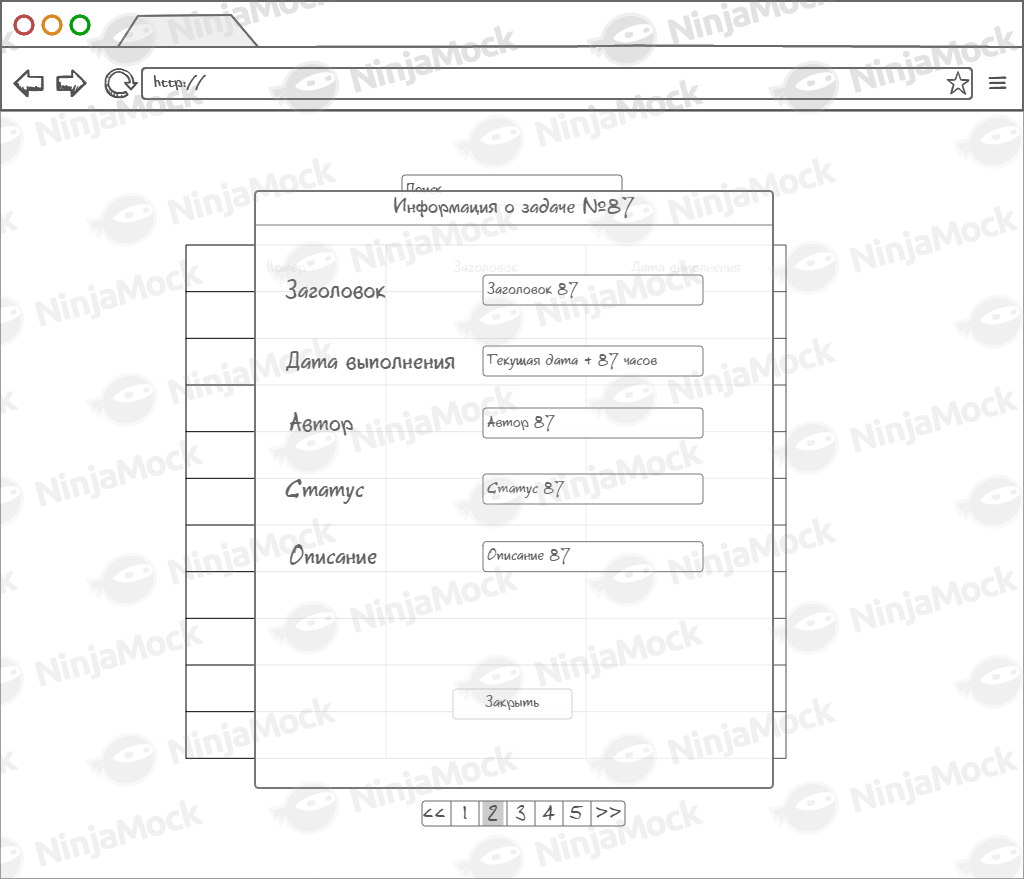


**Дополнительно:** Одновременно должно отображаться 10 задач. Внизу находится пагинатор для переключения между листами задач. Вверху находится строка поиска по полю заголовок по введенной подстроке.

## 

## **Условие 4**

При клике на одну из задач из предыдущего задания должно происходить обращение к веб-сервису получения информации по задаче и отображаться модальное окно с этой информацией. Пример:



**Дополнительно:** запоминать информацию по полученным задачам и при повторном просмотре не делать запрос к веб-сервису.

## 

## **Общие требования**

* База данных может быть любой (SQLite, MariaDB, Postgres и т.п.) Нам важно проверить написание SQL запросы, а не знание СУБД.
* Использование PHP и JS строго без фреймворков.
* CSS тоже без фреймворков и нужен только для отображения всего задания и не более.