

# Привет, я тестовое задание

## Что мы проверяем

Основной целью данного тестового задания является проверка ваших hard skills.

Кроме того, задание покажет соответствие кандидата требованиям из вакансии, а также подтвердит умения:

- творчески подходить к решению задачи,
- ставить дополнительные внутренние задачи самому себе,
- рассматривать поставленную задачу не только с точки зрения разработчика, но и архитектора,
- обеспечивать надежность и качество написанного кода,
- обеспечивать поддерживаемость написанного кода,
- создавать визуально привлекательный продукт.

## Как будем проверять

Будет проведено стандартное в нашей компании code review с выявлением сильных и слабых сторон:

- непосредственно кода,
- принятых архитектурных и алгоритмических решений,
- использованных технологий.

## Требования к заданию

- 1) Выполнение каждого из требований и пунктов задания. Невыполнение хотя бы одного из требований означает невыполнение задания.
- 2) Если в задании не указано явно, что и как сделать - значит реализация оставлена на ваше усмотрение, мы хотим проследить, какие решения вы принимали и как решали поставленную задачу.
- 3) Кандидату предлагается программно реализовать **поставленную задачу** (см. ниже)
- 4) Результат разместить в собственном открытом [github](#)-репозитории и по готовности отправить ссылку на ветку по адресу [job@anychart.com](mailto:job@anychart.com).
- 5) В README репозитория должны быть описаны:
  - а) Обзор сделанного в процессе программной реализации и особенностей, которые вы пожелали добавить;
  - б) Список использованных технологий + краткое обоснование выбора (что использовано и почему);
  - в) Пошаговая инструкция по запуску, от подготовки окружения до визуального отображения в окне браузера;

- d) [опционально] Если вы используете какое-либо тестирование, то требуется также пошаговая инструкция по запуску тестов;
- e) [опционально] Любая дополнительная информация, которую вы сочтете нужной отразить в README.

## Задание

- 1) Пусть имеется небольшой написанный вами на node.js сервер, который по get-запросу отдает данные в формате JSON, следующего вида:

```
{
  ...
  data: [
    { x: string, value: number },
    { x: string, value: number },
    { x: string, value: number },
    ...
  ]
  ...
}
```

- 2) Пример таких данных можно посмотреть [здесь](http://static.anychart.com/cdn/anydata/common/11.json):
- 3) Создайте веб-приложение представляющее собой веб-страницу с полем ввода для URL источника данных (ссылка на API вашего сервера на node) и кнопку “Обновить данные”, при нажатии которой приложение (его DataController, см. пункт 4) получает и обрабатывает данные и отрисовывает их (с помощью его ChartController, см. пункт 5) на странице в виде линейной диаграммы (см пункт 6).

После первичной отрисовки приложение должно с некоторой периодичностью опрашивать источник данных, чтобы проверить, не обновились ли данные и обновлять диаграмму.

- 4) Приложение должно содержать в себе DataController:
- a) DataController - программная сущность, которая получает данные в формате JSON по указанному адресу
  - b) DataController готовит (обрабатывает) данные для удобства визуализации
  - c) DataController умеет настраивать источник данных, то есть, пользователь может указать, по какому URL получать данные,
  - d) Формат данных – JSON вида:

```
{
  ...
  data: [
    { x: string, value: number },
    { x: string, value: number },
    { x: string, value: number },
  ]
}
```

```
...  
]  
...  
}
```

- e) DataController умеет периодически опрашивать источник данных, и если данные изменились, DataController сообщает об этом.
- 5) Приложение должно содержать в себе ChartController:
  - a) ChartController - занимается отрисовкой данных на веб-странице.
  - b) ChartController умеет слышать сообщения от DataController о том, что данные изменились и, соответственно, в зависимости от этого, перерисовывать чарт по новым данным от DataController.
  - c) ChartController рисует линейную диаграмму (см пункт 6) по пришедшим данным.
  - d) Нарисованный график имеет вертикальную и горизонтальную оси
    - i) Оси имеют подписи (на [примере](#) - Jan, Feb, Mar... для горизонтальной оси, 0, 25, 50... для вертикальной)
    - ii) Добавить интерактивность на движение мыши над графиком: должна появляться всплывающая подсказка. Смотреть [пример](#).
- 6) Линейная диаграмма
  - a) Линейная диаграмма или линейный график – это тип диаграммы, которая отображает информацию в виде ряда точек данных, под названием «метки», соединённые прямыми отрезками.
  - b) при выполнении задания можно ориентироваться на пример по ссылке <https://www.anychart.com/chartopedia/chart-type/line-chart/>
- 7) Требования
  - a) График **должен** быть отображен на странице в формате [SVG](#)
  - b) **Нельзя** использовать готовые решения для отрисовки графиков ([Anychart](#), [Highcharts](#), [Recharts](#) и тому подобные)
  - c) Для рендеринга SVG **запрещено** использовать готовые решения типа [d3](#), [GraphicsJs](#), [raphaeljs](#) и даже JQuery.
  - d) Подсказка: все SVG должно быть отрисовано на уровне браузерного API.

## Что дальше

Как только мы получим от вас ссылку на репозиторий – сразу же сообщим, и постараемся как можно быстрее провести code review и сообщить вам о его результатах, и назначить время очного (дистанционного) собеседования.