1. Клонируйте репозиторий https://github.com/womenInIt/restfull\_api
2. установите все пакеты, необходимые для проекта с помощью npm install
3. откройте package.json и найдите команду для запуска проекта в режиме dev, запустите проект, откройте swagger интерфейс, проверьте работают ли конечные точки (end points)

1.часть: получение данных из другого api и передача по запросу ()

|  |  |
| --- | --- |
| получение данных из другого api может происходить по запросу fetch. измените функцию get в файле weather.js и добавьте следующий код:  Измените запрос, так чтобы в пути запроса был параметр town | fetch(`${baseWeatherUrl}?q=${req.params.town}&appid=${apiKey}`)  .then(response => response.json())  .then(data => {  console.log(data);  return res.send(data);  }).catch(err => {  console.log(err);  return res.send(err);  }); |

2.часть: сохранение данных и выдача по запросу (ideas.js)

|  |  |
| --- | --- |
| данные на сервере должны каким либо образом сохраняться. Измените запрос get в файле ideas.js, чтобы получать данные из файла resourses/textdata.json | try {  let rawdata = fs.readFileSync(path.resolve('./resourses/textdata.json'), "utf-8");  let ideas = JSON.parse(rawdata);  res.send(ideas);  } catch (error) {  res.send(error);  } |
| добавьте новую конечную точку точку, и верните цитату по id | router.get('/:id' ….  ideas.find(x => x.id === req.params.id) |
| добавьте пост запрос | router.post('/', function(req, res ){  console.log('request =' + JSON.stringify(req.body))  try {  let rawdata = fs.readFileSync(path.resolve('./resourses/textdata.json'), "utf-8");  let ideas = JSON.parse(rawdata);  ideas.push(req.body);  fs.writeFileSync(path.resolve('./resourses/textdata.json'), JSON.stringify(ideas));  res.send(200);  } catch (error) {  res.send(error);  }  }); |
| Сделайте проверку на повторение id и если такое уже существует, верните ошибку  Сделайте проверку по полю id и если оно не задано, задайте уникальное | |

1. Клонируйте репозиторий https://github.com/womenInIt/restfull\_api
2. установите все пакеты, необходимые для проекта с помощью npm install
3. откройте package.json и найдите команду для запуска проекта в режиме dev, запустите проект, откройте swagger интерфейс, проверьте работают ли конечные точки (end points)

1.часть: получение данных из другого api и передача по запросу ()

|  |  |
| --- | --- |
| получение данных из другого api может происходить по запросу fetch. измените функцию get в файле weather.js и добавьте следующий код:  Измените запрос, так чтобы в пути запроса был параметр town | fetch(`${baseWeatherUrl}?q=${req.params.town}&appid=${apiKey}`)  .then(response => response.json())  .then(data => {  console.log(data);  return res.send(data);  }).catch(err => {  console.log(err);  return res.send(err);  }); |

2.часть: сохранение данных и выдача по запросу (ideas.js)

|  |  |
| --- | --- |
| данные на сервере должны каким либо образом сохраняться. Измените запрос get в файле ideas.js, чтобы получать данные из файла resourses/textdata.json | try {  let rawdata = fs.readFileSync(path.resolve('./resourses/textdata.json'), "utf-8");  let ideas = JSON.parse(rawdata);  res.send(ideas);  } catch (error) {  res.send(error);  } |
| добавьте новую конечную точку точку, и верните цитату по id | router.get('/:id' ….  ideas.find(x => x.id === req.params.id) |
| добавьте пост запрос | router.post('/', function(req, res ){  console.log('request =' + JSON.stringify(req.body))  try {  let rawdata = fs.readFileSync(path.resolve('./resourses/textdata.json'), "utf-8");  let ideas = JSON.parse(rawdata);  ideas.push(req.body);  fs.writeFileSync(path.resolve('./resourses/textdata.json'), JSON.stringify(ideas));  res.send(200);  } catch (error) {  res.send(error);  }  }); |
| Сделайте проверку на повторение id и если такое уже существует, верните ошибку  Сделайте проверку по полю id и если оно не задано, задайте уникальное | |