**ЛАБОРАТОРНАЯ РАБОТА №9**

**АВТОРИЗАЦИЯ ПО ПРОТОКОЛУ OAUTH**

**(SPRING BOOT)**

Цель работы: получение навыков разработки системы авторизации по протоколу OAuth.

Программное обеспечение: IntelliJ IDEA, JDK 1.8+, Spring Boot, Hibernate, MySQL.

Необходимая теоретическая подготовка: Maven, Java Spring, Spring Boot, OAuth.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Google

На странице входа в систему нашего сайта пользователи могут выбрать один из двух вариантов входа. Первый — указать имя пользователя и пароль, если у них уже есть учетная запись на нашем сайте. А другой заключается в том, чтобы войти на сайт с помощью существующей учетной записи Google.

Когда они нажимают кнопку «Войти через Google», запускается поток авторизации через Google и он переводит пользователей на сайт Google. Оказавшись там, они входят в систему со своими учетными данными Google, после чего они будут перенаправлены на страницу подтверждения.

На странице подтверждения у пользователей будет запрошено разрешение на передачу информации о своей учетной записи Google на сторонний сайт. В этом случае сторонний сайт — это сайт, на котором они хотят использовать свою учетную запись Google для входа в систему. Им будут представлены два варианта: они могут разрешить или запретить это.

После того, как они разрешат поделиться своей информацией со сторонним сайтом, пользователи будут перенаправлены на сторонний сайт, откуда они инициировали поток входа в Google.

На этом этапе пользователь вошел в систему с помощью Google, а сторонний сайт имеет доступ к информации профиля пользователя, которую можно использовать для создания учетной записи и входа в систему.

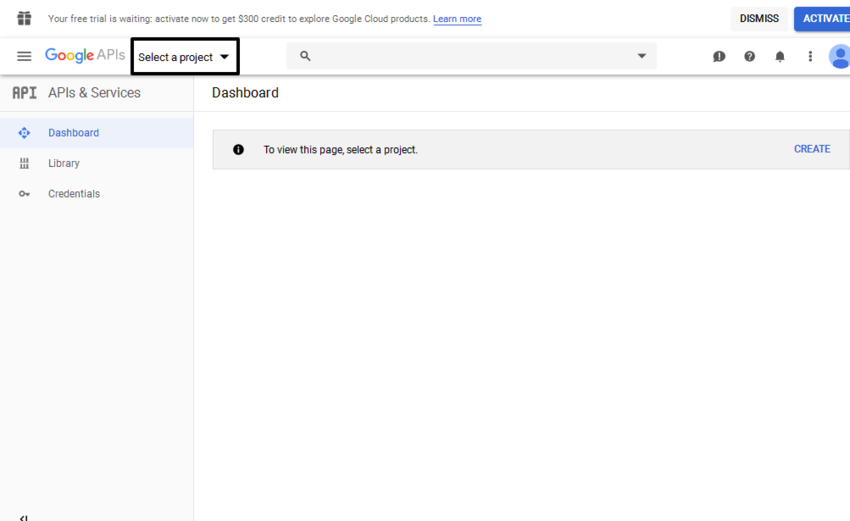
Это основной процесс интеграции авторизации Google на нашем сайте.

Настройка проекта для авторизации через Google

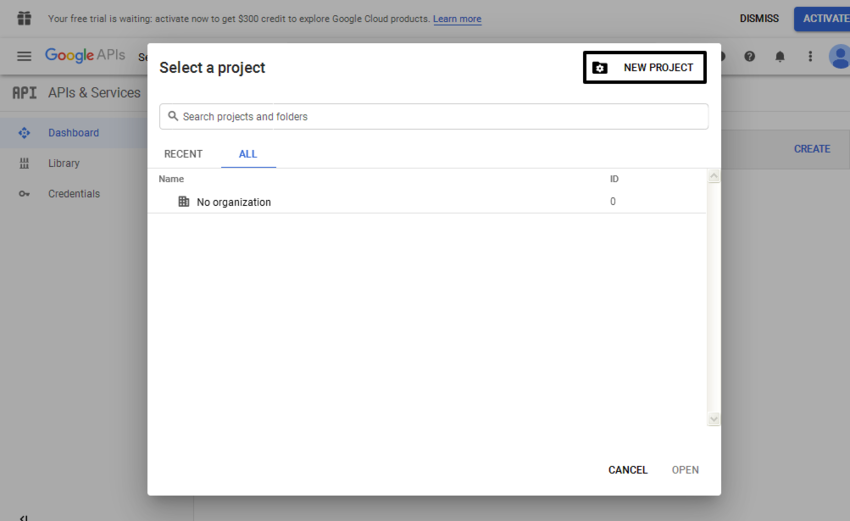
Создайте проект Google API

Во-первых, вам нужно создать приложение в Google, которое позволит вам зарегистрировать свой сайт в Google. Это позволяет вам настроить основную информацию о вашем сайте, а также пару технических деталей.

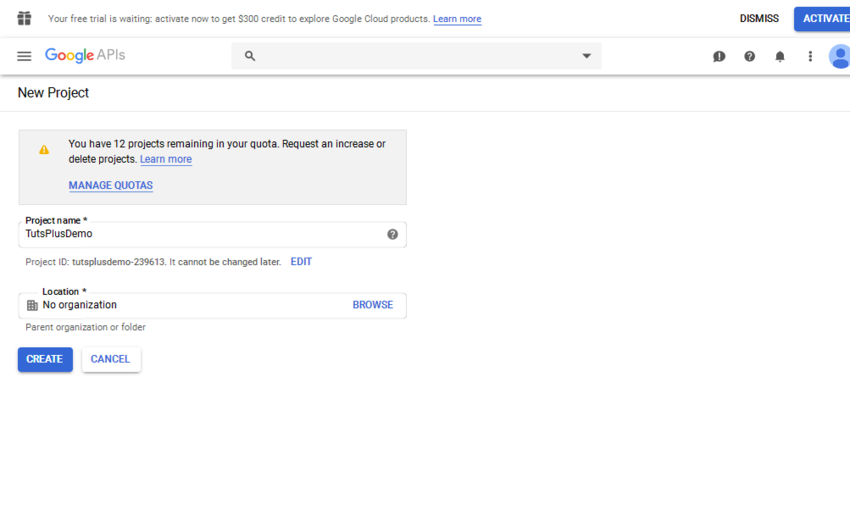
После того, как вы вошли в систему с помощью Google, откройте [консоль разработчиков Google](https://console.developers.google.com/). Откроется страница панели инструментов Google, как показано на следующем снимке экрана.



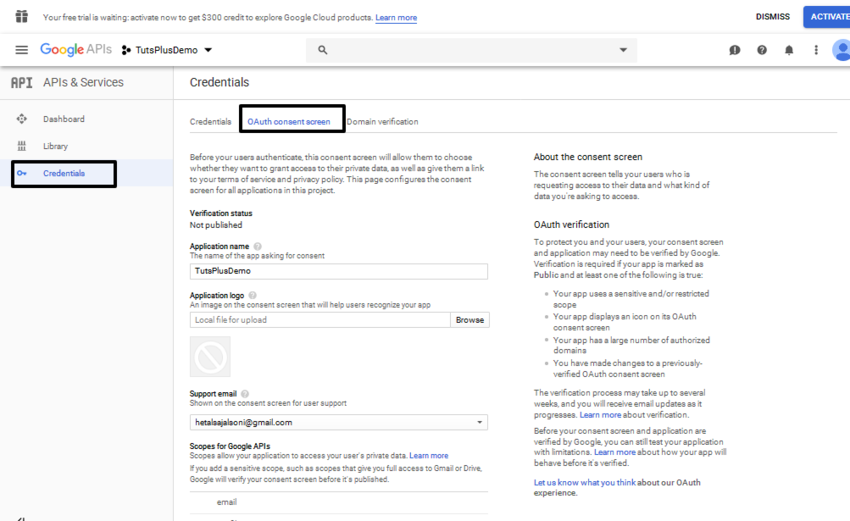
В верхнем левом меню нажмите на ссылку «Выбрать проект». Откроется всплывающее окно, как показано на снимке экрана.



Нажмите на ссылку «Новый проект», в появившейся панели вам нужно будет ввести название проекта и другую информацию. Заполните необходимые данные, как показано в следующем снимке экрана.

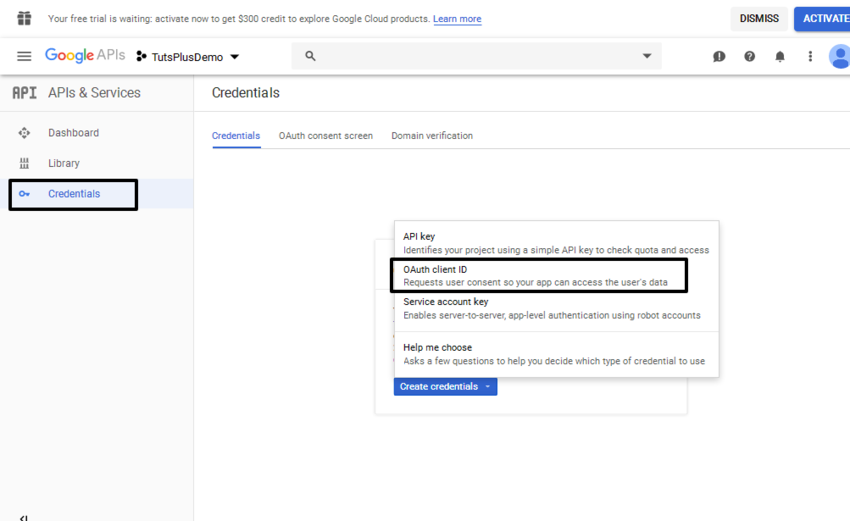


Нажмите кнопку «Создать», чтобы сохранить новый проект. Вы будете перенаправлены на страницу панели инструментов. Нажмите «Учетные данные» в левой боковой панели и перейдите на вкладку настроек OAuth.

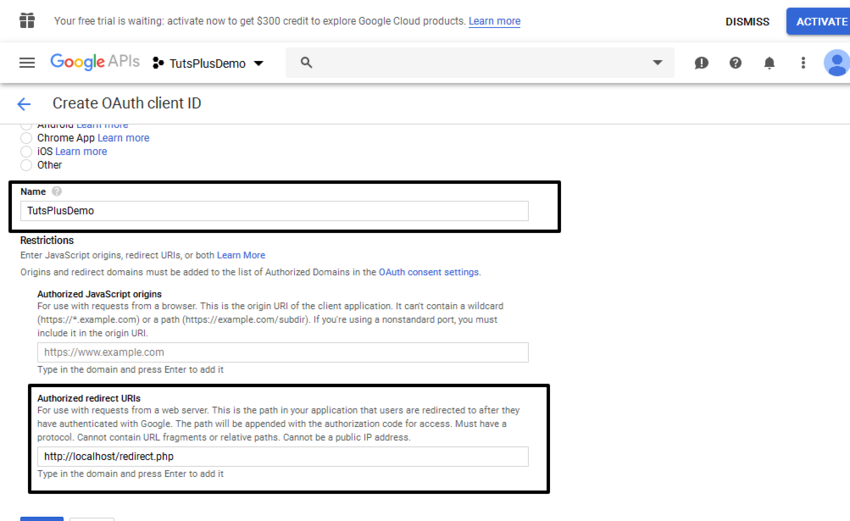


На этой странице вам нужно ввести информацию о приложении, такую как название приложения, логотип и некоторые другие данные. Заполните необходимые данные и сохраните их. Для целей тестирования достаточно просто ввести имя приложения.

Затем нажмите «Учетные данные» в левой боковой панели. Должно отобразиться поле «Учетные данные API», как показано на следующем снимке экрана.



Нажмите «Учетные данные клиента»> «Идентификатор клиента OAuth», чтобы создать новый набор учетных данных для приложения. Отобразится панель, в которой вам будет предложено выбрать нужный вариант. В нашем случае выберите опцию «Веб-приложение» и нажмите кнопку «Создать». Вам будет предложено предоставить несколько более подробную информацию.



Введите данные, показанные на снимке экрана выше, и сохраните их! Конечно, вам нужно установить URI перенаправления в соответствии с настройками вашего приложения. Это URL, по которому пользователь будет перенаправлен после входа в систему.

На данный момент мы создали клиентское приложение Google OAuth2, и теперь нам нужно получить возможность использовать это приложение для интеграции на нашем сайте входа в систему через Google. Пожалуйста, запишите идентификатор клиента и секретный ключ клиента, они потребуются при настройке приложения с нашей стороны. Вы всегда можете найти идентификатор клиента и секретный ключ клиента при редактировании приложения.

Установите клиентскую библиотеку Google PHP SDK

Есть два варианта установки. Использовать Composer.

Скачать релиз

Если вы не хотите использовать Composer, вы также можете загрузить последнюю стабильную версию с [официальной страницы API](https://github.com/googleapis/google-api-php-client/releases). В моем примере я просто использовал Composer.

К настоящему момент у вас должно быть настроено приложение Google и установлена клиентская библиотека API Google PHP. В следующем и последнем разделе мы рассмотрим, как использовать эту библиотеку на сайте.

Интеграция клиентской библиотеки

Напомним, что при настройке приложения Google мы предоставили URI перенаправления в конфигурации приложения. Мы задали адрес http: //localhost/redirect.php. Теперь пришло время создать файл redirect.php. Создайте файл redirect.php со следующим содержимым.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | <?php  require\_once 'vendor/autoload.php';    // init configuration  $clientID = '<YOUR\_CLIENT\_ID>';  $clientSecret = '<YOUR\_CLIENT\_SECRET>';  $redirectUri = '<REDIRECT\_URI>';    // create Client Request to access Google API  $client = new Google\_Client();  $client->setClientId($clientID);  $client->setClientSecret($clientSecret);  $client->setRedirectUri($redirectUri);  $client->addScope("email");  $client->addScope("profile");    // authenticate code from Google OAuth Flow  if (isset($\_GET['code'])) {    $token = $client->fetchAccessTokenWithAuthCode($\_GET['code']);    $client->setAccessToken($token['access\_token']);      // get profile info    $google\_oauth = new Google\_Service\_Oauth2($client);    $google\_account\_info = $google\_oauth->userinfo->get();    $email =  $google\_account\_info->email;    $name =  $google\_account\_info->name;      // now you can use this profile info to create account in your website and make user logged in.  } else {    echo "<a href='".$client->createAuthUrl()."'>Google Login</a>";  }  ?> |

Давайте пройдемся по ключевым частям кода. Первое, что нам нужно сделать, это включить файл autoload.php. Это часть Composer, она гарантирует, что классы, которые мы используем в скрипте, загружаются автоматически.

Далее у нас есть раздел конфигурации, который инициализирует конфигурацию приложения путем установки необходимых настроек. Конечно, вам необходимо заменить заполнители на соответствующие значения.

|  |  |
| --- | --- |
| 1  2  3  4 | // init configuration  $clientID = '<YOUR\_CLIENT\_ID>';  $clientSecret = '<YOUR\_CLIENT\_SECRET>';  $redirectUri = '<REDIRECT\_URI>'; |

В следующем разделе создается экземпляр объекта Google\_Client, который будет использоваться для выполнения различных действий. Наряду с этим мы также инициализировали настройки нашего приложения.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // create Client Request to access Google API  $client = new Google\_Client();  $client->setClientId($clientID);  $client->setClientSecret($clientSecret);  $client->setRedirectUri($redirectUri); |

Затем мы добавили электронную почту и данные профиля, таким образом после входа в систему у нас есть доступ к основной информации профиля.

|  |  |
| --- | --- |
| 1  2 | $client->addScope("email");  $client->addScope("profile"); |

Наконец, у нас есть фрагмент кода, который выполняет магию потока входа в систему.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | // authenticate code from Google OAuth Flow  if (isset($\_GET['code'])) {    $token = $client->fetchAccessTokenWithAuthCode($\_GET['code']);    $client->setAccessToken($token['access\_token']);      // get profile info    $google\_oauth = new Google\_Service\_Oauth2($client);    $google\_account\_info = $google\_oauth->userinfo->get();    $email =  $google\_account\_info->email;    $name =  $google\_account\_info->name;      // now you can use this profile info to create account in your website and make user logged in.  } else {    echo "<a href='".$client->createAuthUrl()."'>Google Login</a>";  } |

Во-первых, часть else, которая будет запущена при непосредственном доступе к скрипту. Она отображает ссылку, которая ведет пользователя в Google для входа в систему. Важно отметить, что мы использовали метод createAuthUrl из Google\_Client для создания URL OAuth.

После нажатия на ссылку для авторизации через Google пользователи будут перенаправлены на сайт Google. После входа в систему Google перенаправляет пользователей на наш сайт, передавая переменную строки запроса code. И вот тогда будет запущен код PHP в блоке if. Мы будем использовать код для обмена токеном доступа.

Получив токен доступа, мы можем использовать сервис Google\_Service\_OАuth2 для получения информации о профиле вошедшего в систему пользователя.

Таким образом, мы получим доступ к информации профиля, как только пользователь войдет в учетную запись Google. Мы можем использовать эту информацию для создания учетных записей на своем сайте или сохранить ее в сеансе.