# **Лабораторная работа №3**

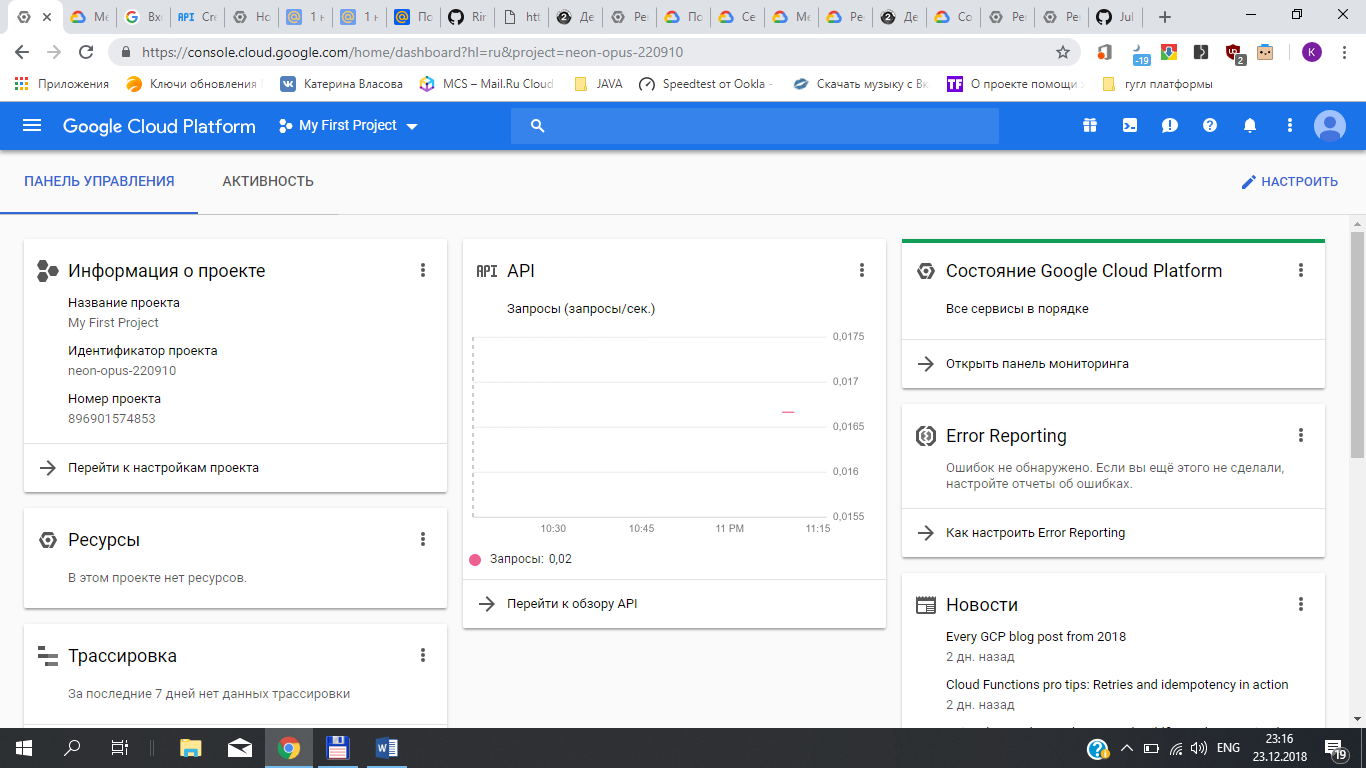
**Задание:** реализовать веб-приложение с минимальным пользовательским интерфейсом, которое позволяет управлять созданием\удалением пользователей в  google cloud platfom.

**Цель:** получить навыки работы с протоколом управления пользователями и их полномочиями в google cloud platform.

**Ход работы:**

**Ход работы:**

1. Регистрируемся в Google  Cloud Platform.
2. Создаём проект.



1. Создаём API ключ. Ключ API - это простая зашифрованная строка, которую можно использовать при вызове определенных API, которым не требуется доступ к частным данным пользователя. Ключи API полезны в клиентах, таких как браузер и мобильные приложения, которые не имеют внутреннего сервера. Ключ API используется для отслеживания запросов API, связанных с вашим проектом, для квоты и выставления счетов.

Ключи API имеют важные ограничения, такие как:

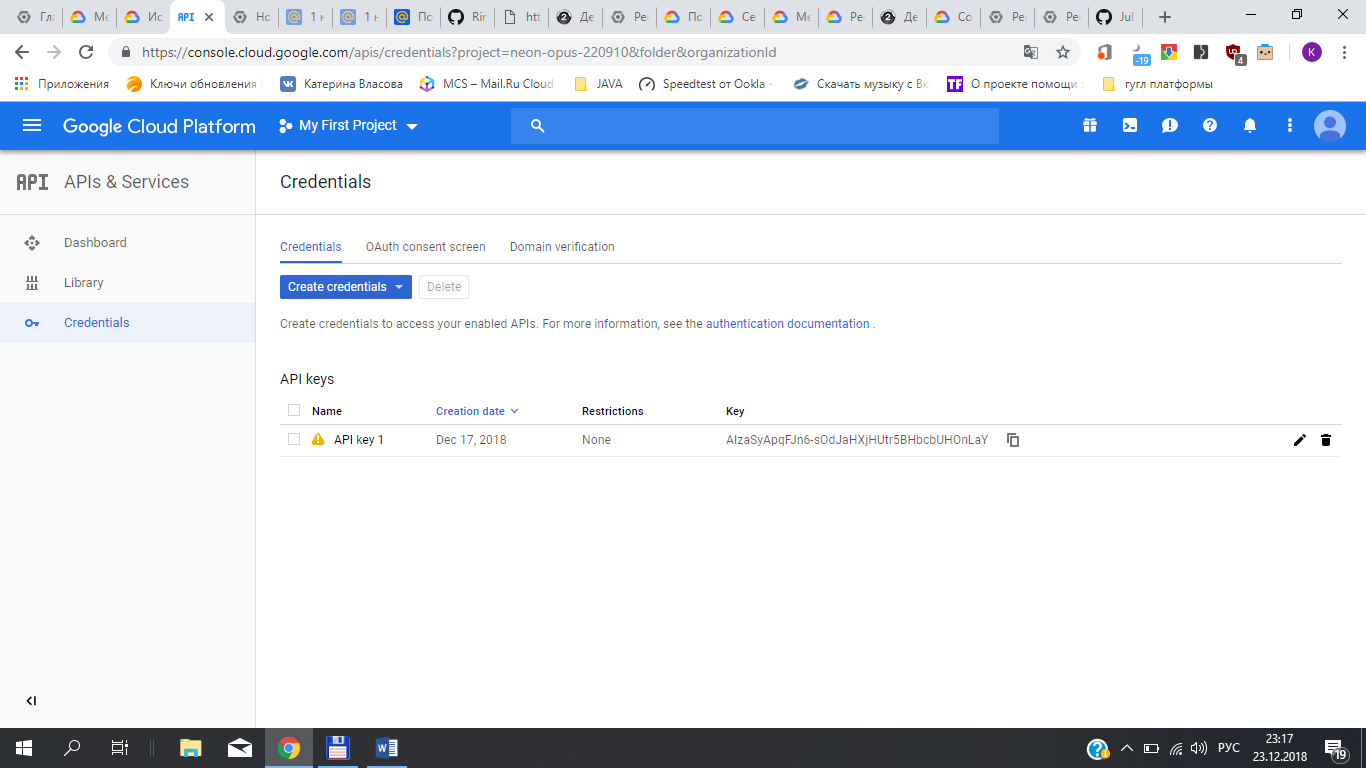
* Ограниченное количество сервисов GCP разрешает доступ, используя только ключ API:

[API Google Cloud Natural Language](https://cloud.google.com/natural-language/), [Google Cloud Speech API](https://cloud.google.com/speech/), [API Google Cloud Translation](https://cloud.google.com/translate/), [API Google Cloud Vision](https://cloud.google.com/vision/), [Конечные точки Google Cloud](https://cloud.google.com/endpoints/), [API Google Billing Catalog API](https://cloud.google.com/billing/), [API предотвращения потери данных в облаке](https://cloud.google.com/dlp/)

* Ключи API не идентифицируют пользователя или приложение, отправляющее запрос API, поэтому вы не можете ограничить доступ к определенным пользователям или учетным записям служб.
* Другим проще обнаружить и использовать ваш ключ API.

Чтобы создать ключ API, вашей учетной записи должна быть предоставлена ​​примитивная роль редактора ( roles/editor) в текущем проекте.  Чтобы создать ключ API:

* 1. Перейдите к панели [API & Services → Credentials](https://console.cloud.google.com/apis/credentials) в консоли GCP.
  2. Выберите Создать учетные данные , затем выберите ключ API в раскрывающемся меню.
  3. Нажмите кнопку **Создать** . В диалоговом окне Созданный ключ **API** отображается ваш вновь созданный ключ.



1. Для передачи ключа API в API REST в качестве параметра запроса в следующем формате. Замените API\_KEY своим ключом API: *KEY* = *API\_KEY.*
2. Получаем код авторизации Exchange для токенов , тем самым появится обновление и токен доступа, необходимый для доступа к защищенным ресурсам OAuth.
3. Создаем HTTP-запрос:

POST [https://iam.googleapis.com/v1/{name=projects/\*}/serviceAccounts](https://iam.googleapis.com/v1/%7bname=projects/*%7d/serviceAccounts),

где: name - имя ресурса проекта, типа string

Тело запроса содержит данные со следующей структурой:

{ "accountId" : string,

"serviceAccount" : {

object ( [ServiceAccount](https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts#ServiceAccount)) } },

где:

- accountId - идентификатор учетной записи, который используется для создания адреса электронной почты учетной записи службы и стабильного уникального идентификатора, типа string

- serviceAccount – ресурс для создания, типа object

1. Инициируем HTTP-запрос отправив этот самый запрос. Для того что бы удалять и добавлять пользователей не через консоль, в Javascript добавляем код. Например:

*<script src="https://apis.google.com/js/api.js"></script>*

*<script>*

*/\*\**

*\* Sample JavaScript code for iam.roles.list*

*\* See instructions for running APIs Explorer code samples locally:*

*\* https://developers.google.com/explorer-help/guides/code\_samples#javascript*

*\*/*

*function authenticate() {*

*return gapi.auth2.getAuthInstance()*

*.signIn({scope: "https://www.googleapis.com/auth/cloud-platform"})*

*.then(function() { console.log("Sign-in successful"); },*

*function(err) { console.error("Error signing in", err); });*

*}*

*function loadClient() {*

*return gapi.client.load("https://content.googleapis.com/discovery/v1/apis/iam/v1/rest")*

*.then(function() { console.log("GAPI client loaded for API"); },*

*function(err) { console.error("Error loading GAPI client for API", err); });*

*}*

*// Make sure the client is loaded and sign-in is complete before calling this method.*

*function execute() {*

*return gapi.client.iam.roles.list({})*

*.then(function(response) {*

*// Handle the results here (response.result has the parsed body).*

*console.log("Response", response);*

*},*

*function(err) { console.error("Execute error", err); });*

*}*

*gapi.load("client:auth2", function() {*

*gapi.auth2.init({client\_id: YOUR\_CLIENT\_ID});*

*});*

*</script>*

*<button onclick="authenticate().then(loadClient)">authorize and load</button>*

*<button onclick="execute()">execute</button>*

*{*

*"roles": [*

*{*

*"name": "roles/accesscontextmanager.policyAdmin",*

*"title": "Access Context Manager Admin",*

*"description": "Full access to policies, access levels, and access zones",*

*"stage": "BETA",*

*"etag": "AA=="*

*},*

*{*

*"name": "roles/accesscontextmanager.policyEditor",*

*"title": "Access Context Manager Editor",*

*"description": "Edit access to policies. Create, edit, and change access levels and access zones.",*

*"stage": "BETA",*

*"etag": "AA=="*

*},*

*{*

*"name": "roles/accesscontextmanager.policyReader",*

*"title": "Access Context Manager Reader",*

*"description": "Read access to policies, access levels, and access zones.",*

*"stage": "BETA",*

*"etag": "AA=="*

*},*

*{*

*"name": "roles/androidmanagement.user",*

*"title": "Android Management User",*

*"description": "Full access to manage devices.",*

*"stage": "GA",*

*"etag": "AA=="*

*},*

*{*

*"name": "roles/appengine.appViewer",*

*"title": "App Engine Viewer",*

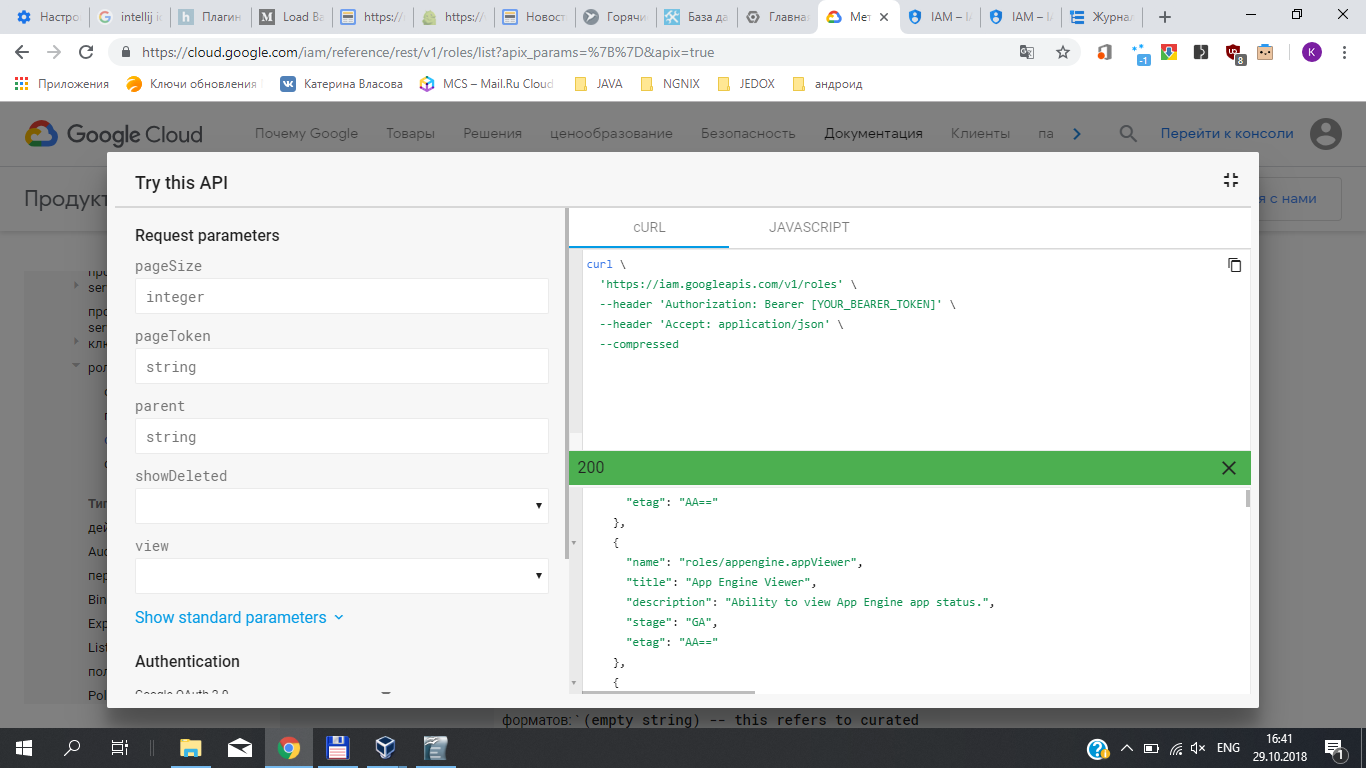
*"description": "Ability to view App Engine app status.",*

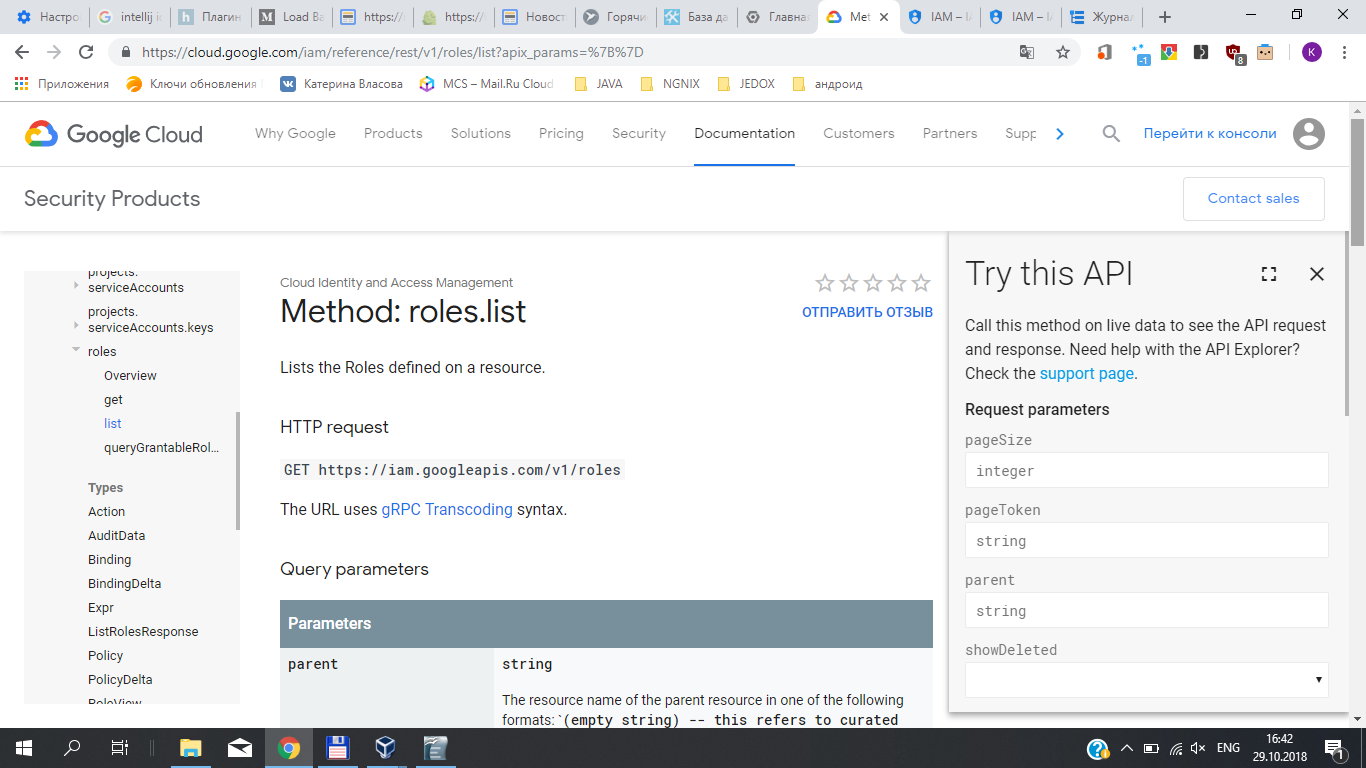
*"stage": "GA",*

*"etag": "AA=="*

*},*

*…......*

****

****