

Sveučilište u Zagrebu  
Fakultet organizacije i informatike,  
Pavlinska ulica 2,  
42000 Varaždin, Hrvatska

## **DNEVNIK STRUČNE PRAKSE**

### **Osnovni podaci o studentu/ici:**

|                        |                                  |
|------------------------|----------------------------------|
| <b>Ime i prezime:</b>  | Matej Desanić                    |
| <b>JMBAG broj:</b>     | 0016155191                       |
| <b>Akadska godina:</b> | 3.                               |
| <b>Studij:</b>         | Informacijski i poslovni sustavi |

### **Osnovni podaci o poduzeću:**

|                                  |                            |
|----------------------------------|----------------------------|
| <b>Naziv tvrtke/institucije:</b> | FEROMIHIN                  |
| <b>Adresa:</b>                   | Samoborska ulica 15        |
| <b>Kontakt:</b>                  | luka.finderle@asynclabs.co |
| <b>Mentor/ica:</b>               | Luka Finderle              |
| <b>Datum početka prakse:</b>     | 1.7.2024.                  |
| <b>Datum završetka prakse:</b>   | 5.7.2024.                  |
| <b>Ukupno odrađen fond sati:</b> | 40                         |

**Student/ica je obavio/la stručnu praksu u trajanju od \_\_\_\_\_ dana i redovito vodio/la dnevnik stručne prakse te time izvršio/la obvezu stručne prakse na kolegiju**

\_\_\_\_\_.

**Zapažanje mentorice:**

---

---

---

---

---

---

---

---

---

\_\_\_\_\_  
(Potpis mentora/ice)

**U Zagrebu, \_\_\_\_\_**

## 1. Dan – 1.7.2024.

Radni dan je počeo internim sastankom između mene i mentora Luke Finderle u konferencijskoj sobi u sklopu objekta u kojem se nalazi tvrtka FEROMIHIN na adresi Samoborska ulica 15. Mentor me uveo u kulturu tvrtke, upoznao me sa postojećim odjelima u tvrtci koji su vezani za razvoj i isporuku programskih proizvoda kao što su frontend, backend, QA i DevOps. Tvrtka je fleksibilna oko odabira metodologija i pristupa pri planiranju, dizajnu, razvoju i isporuci proizvoda. Voditelji timova kombiniraju aspekte različitih pristupa, poput kombiniranja vodopadnog pristupa u stilu analiziranja i dizajniranja rješenja, te se onda prilikom software developmenta prebacuju na agilan razvoj sa sprintovima. Za međusobnu komunikaciju radnici koriste alat Slack, te su svi međusobno spojeni na kalendare kako bi korektno mogli pratiti svoje obaveze i pratiti što se dešava u tvrtci u datom momentu, te sam i ja koristio Slack za komunikaciju sa svojim mentorom.

Mentor me također upoznao s mojim tjednim zadatkom, a to je istraživanje mogućnosti integracije AutoCAD-a s web sučeljem te implementacija integracije neke odabrane biblioteke u web sučelju. Ideja je da rad prezentiram timu sa zaključcima o izvedivosti i potencijalnim izazovima. Za rad je bilo potrebno urediti radno okruženje. Pripremio sam Node.js, Git, Visual Studio Code, no taj proces je bio jednostavan jer sam već imao sve spremno na računalu, samo je trebalo ažurirati potencijalno zastarjele verzije software-a. Kako se dosad nisam susretao s React okruženjem, za postavljanje njega sam proučio ovaj video <https://www.youtube.com/watch?v=HldPpm-0ZNO>. Nakon malih problema oko kojih mi je kolega pomogao jer poznaje sintaksu za baratanje putanjama i konfiguriranje okruženja dok ja još nisam dovoljno vješt u tome, uspješno sam postavio React okruženje. Ispod sam prikazao log terminala prilikom instalacije React okruženja. Nakon toga, trebao sam istražiti AutoCAD API i dokumentaciju što je mi je uzelo većinu radnog dana jer dosad nisam bio upoznat s AutoCAD-om. Kako pišem ispit iz Testiranja i kvalitete proizvoda uspjeli smo se dogovoriti da odradim danas i sutra 10 sati, te da u srijedu dođem nakon ispita odraditi 4 sata. U prilogu se nalazi screenshot Slack-a gdje samo dobio zadatak za ovih 5 dana.

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npx create-react-app test
```

```
npm ERR! syscall Istat
```

```
npm ERR! errno -4058
```

```
npm ERR! enoent ENOENT: no such file or directory, Istat 'C:\Users\Geoff\AppData\Roaming\npm'
```

```
npm ERR! enoent This is related to npm not being able to find a file.
```

```
npm ERR! enoent
```

```
npm ERR! A complete log of this run can be found in: C:\Users\Geoff\AppData\Local\npm-cache\_logs\2024-07-04T08_41_48_159Z-debug-0.log
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npm -v
```

```
10.5.2
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> mkdir -p C:\Users\Geoff\AppData\Roaming\npm
```

```
-----  
d----- 7/4/2024 10:42 AM      npm
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npm config set prefix  
"C:\Users\Geoff\AppData\Roaming\npm"
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npm config set cache  
"C:\Users\Geoff\AppData\Local\npm-cache"
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npm cache clean --force
```

```
npm WARN using --force Recommended protections disabled.
```

```
PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npm install -g create-react-app
```

```
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
```

```
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
```

```
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
```

```
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check  
out lru-cache if you want a good and tenpm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no  
longer supported
```

```
npm WARN deprecated fstream@1.0.12: This package is no longer supported.
```

```
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security  
updates. Please upgrade asap.
```

added 64 packages in 4s

4 packages are looking for funding

run `npm fund` for details

PS C:\Users\Geoff\Documents\Praksa\Software\ReactJS> npx create-react-app test

Creating a new React app in C:\Users\Geoff\Documents\Praksa\Software\ReactJS\test.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts with cra-template...

added 1482 packages in 1m

262 packages are looking for funding

run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 63 packages, and changed 1 package in 9s

262 packages are looking for funding

run `npm fund` for details

Removing template package using npm...

removed 1 package, and audited 1545 packages in 4s

262 packages are looking for funding

run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:

```
npm audit fix --force
```

Run `npm audit` for details.

Created git commit.

Success! Created test at C:\Users\Geoff\Documents\Praksa\Software\ReactJS\test

Inside that directory, you can run several commands:

```
npm start
```

Starts the development server.

```
npm run build
```

Bundles the app into static files for production.

```
npm test
```

Starts the test runner.

```
npm run eject
```

and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

```
cd test
```

```
npm start
```

Happy hacking!



## 2. Dan – 2.7.2024.

Danas sam odlučio maksimalno se posvetiti razumijevanju svakog od prije prikazanih AutoCAD API-ja, jer sam jučer već uspio shvatiti koji točno API-ji imaju mogućnost integracije s web sučeljem. O svakom API-ju sam definirao tri glavne stavke zbog kojih se koristi kako bih mogao kvalitetno predstaviti kolegama koje mogućnosti postoje. Kad sam naučio i zapisao sve opcije, komunicirao sam s mentorom o tome za što im točno trebaju AutoCAD API-ji, odnosno koji je daljnji plan s razvojem nakon što prezentiram Proof of Concept, kako bih mogao odabrati adekvatan API za planiranje i izradu Proof of Concept-a.

Odlučili smo se kako bi najpametnije bilo prvo implementirati Authentication API, jer bez njega ostali API-ji ionako nemaju funkcionalnost. Authentication API koristi OAuth 2.0 standard za generiranje sigurnosnih tokena koji omogućavaju autentifikaciju zahtjeva prema APS API-jima i SDK-ovima. Ovo je ključan korak jer osigurava da samo ovlašteni korisnici mogu pristupiti i koristiti druge API-je. Korištenje ovog API-ja omogućava sigurnu i pouzdanu autentifikaciju korisnika, što je temelj za daljnje razvojne aktivnosti.

Također, zapisao sam sve što sam naučio u vlastitu dokumentaciju, koju ću koristiti prilikom izrade prezentacije za tim. Osigurao sam da dokumentacija bude jasna i sažeta, kako bi kolege mogli brzo shvatiti glavne prednosti i mogućnosti svakog API-ja.

Ovaj prvi korak, iako osnovni, postaviti će temelj za daljnju integraciju ostalih API-ja, omogućujući nam da sigurno pristupamo i manipuliramo podacima unutar AutoCAD okruženja. Sljedeći korak bit će izrada Proof of Concept-a koji će demonstrirati kako različiti API-ji mogu biti integrirani u stvarnoj aplikaciji, što će omogućiti daljnju integraciju s ostalim funkcionalnostima AutoCAD-a. U prilogu se nalazi dio dokumenta koji ću koristiti za izradu prezentacije, te web stranice koje sam posjetio.

The screenshot shows a Notepad application window titled 'Istraživanje AutoCAD API-ja'. The main content area lists several APIs and their capabilities:

- BIM 360 API**
  - Proširenje mogućnosti BIM 360: Integrirajte se s Autodesk BIM 360 za doseganje neiskorištenih segmenata građevinskog ekosustava koji trebaju izravan pristup BIM podacima.
  - Pristup podacima: Pristupajte podacima iz BIM 360 Teama, Fusion Teama, BIM 360 Docs i Object Storage Servicea.
  - Povećanje vrijednosti: Kreirajte aplikacije koje poboljšavaju prikaz podataka i dodaju vrijednost korisnicima.
- Data Exchange API**
  - Učinkovita povezanost: Efikasno povežite i pristupite podskupovima dizajnerskih podataka u svojoj aplikaciji po izboru, bez obzira na velike prijenose podataka, uvoze ili izvoze.
  - Interoperabilnost: Omogućava razmjenu podataka između različitih aplikacija i sustava.
  - Fleksibilnost: Podržava različite formate podataka i protokole.
- Data Management API**
  - Pristup podacima: Pristup podacima iz BIM 360 Teama, Fusion Teama, BIM 360 Docs i Object Storage Servicea.
  - Upravljanje podacima: Kreirajte aplikacije koje poboljšavaju prikaz podataka i dodaju vrijednost korisnicima.
  - Pohrana podataka: Sigurna pohrana i upravljanje velikim količinama podataka.
- Flow Graph Engine API**
  - Poboljšanje dizajnerskih procesa: Poboljšajte dizajnerske procese i izvodite više operacija istovremeno uz automatizaciju temeljenu na oblaku.
  - Optimizacija resursa: Optimizirajte resurse i vrijeme.
  - Automatizacija: Omogućava automatizaciju složenih radnih tokova.
- Tandem Data API**
  - Pristup digitalnim bilancima: Pristupite svojim digitalnim bilancima, prostorima, sustavima i vremenskim nizovima senzorskih podataka.
  - Integracija: Integrirajte se s vanjskim sustavima za upravljanje održavanjem.
  - Upravljanje podacima: Omogućava upravljanje i analizu podataka u stvarnom vremenu.
- Webhooks API**
  - Notifikacije u stvarnom vremenu: Primajte obavijesti o promjenama na AutoCAD datotekama u stvarnom vremenu.

The right sidebar shows a search bar and a list of tags: autorizaciju, Autodesk Construction Cloud API-ji, Integracijske mogućnosti, Upravljanje podacima, Suradnja, BIM 360 API-ji, Data Exchange API, Data Management API, Flow Graph Engine API.



## Enable secure and seamless data access for applications

Developer guide

## Fundamentals

If you're looking to explore the API further, here are some basic resources to get started:

## API terminology and concepts

Get familiar with the basic flow for using OAuth on Autodesk Platform Services (APS).

➔ **Get the basics**

## Application types



























Learn about the types of applications you can register with APS and their corresponding grant types to access tokens

[➔ View guide](#)





























## Tutorials

Get step-by-step guidance to learn how you can complete the most common API workflows.

| <div> <div>Authentication (OAuth)</div> <div>Version 2</div> </div> <div> <div>Developer's Guide</div> <div>Overview</div> <div>API Basics</div> <div>Application Types</div> <div>Field Guide</div> <div>Scopes</div> <div>Asymmetric Signing</div> <div>Error Handling</div> <div>Rate Limits</div> </div> <div> <div>Step-by-Step Tutorials</div> <div>API Reference</div> </div> | <div> <div>While the steps may get more complex, the basic flow for using OAuth on the platform is as follows:</div> <ol style="list-style-type: none"> <li>1. Your app makes an HTTP call to an OAuth REST endpoint and provides its credentials.</li> <li>2. A token is returned to your app.</li> <li>3. In making subsequent HTTP calls to various APIs on the platform, your app includes the token in a request header.</li> </ol> <div>There are a number of concepts that you need to be familiar with:</div> </div>   |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
|--|--|------|---------|-----|--|-----------|---|---------------|--|---------------------------|---|-----------------------------|--|--------------|---|
|  | <table> <thead> <tr> <th>Term</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>app</td><td> <p>The "app" is the code that you're writing that calls APIs. It can be a web application, a headless server process, a mobile app, or even embedded firmware. Within the AWS developer portal, your app is registered in the My Apps section.</p> <p>Sometimes, "app" refers to your code, other times, it refers to the entity registered in the Dev Portal. It will be clear from context which is meant.</p> <p>When you register your app in the AWS developer portal, you select the APIs that you want to bind to. This ensures that even if you lose control of your authentication credentials, calls cannot be made on behalf to APIs the app was not designed for.</p> <p>A registered app has assigned to it credentials called a "client ID" and "client secret", which are analogous to a username and password, except that you do not select them. These are used to obtain access tokens.</p> </td></tr> <tr> <td>client ID</td><td> <p>The "client ID" is essentially your app's username and can be found in the My Apps section. It is an alphanumeric string (e.g., 5zw9v0aUwM7uX3sc0g2DkVvYXt). and it is passed as the value for the client_id query parameter and JGSA attribute.</p> <p>On other platforms, this is sometimes called "consumer key" or "API key"</p> </td></tr> <tr> <td>client secret</td><td> <p>The "client secret" is essentially your app's password and can be found next to the client ID in the My Apps section. It is an alphanumeric string (e.g., 756uN3p9vexHf0), and it is passed as the value for the client_secret query parameter.</p> <p>On other platforms, this is sometimes called "consumer secret" or "API secret"</p> </td></tr> <tr> <td>two-legged authentication</td><td> <p>This is often considered the default type of authentication, and it is the simplest. When your app needs to call the API without having to access resources that require an end-user's permission. For example, translating a design from one format to another, the communication is directly between your app and AWS. The end user, if there is one, does not need to be aware of the authentication or provide authorization for the app to access any resources.</p> <p>This kind of flow is called "two-legged" because your app and AWS are the two "legs".</p> <p>See the section below about common authentication flows.</p> </td></tr> <tr> <td>three-legged authentication</td><td> <p>When an app needs to access resources belonging to an end user, that user must explicitly provide authorization. The typical mechanism, in the case of a web app, is for the app to redirect the user to an Autodesk login page with information about which of the user's resources it wishes to access. The user then provides explicit consent, letting APIs know that the referring app is to be granted the requested access. The user is then redirected back to the app, which can then access the requested resources.</p> <p>This kind of flow is called "three-legged" because your app, AWS, and the end user make up the three "legs" of the flow. Three-legged authentication can be done by using authorization code. When the authorization code grant is used, an Authorization Code is issued to the user to exchange it to obtain Access Token and Refresh Token.</p> <p>See the section below about common authentication flows.</p> </td></tr> <tr> <td>callback URL</td><td> <p>In certain contexts, your app needs to hand over part of the authentication flow to AWS. For example, in three-legged authentication for</p> </td></tr> </tbody> </table> | Term | Meaning | app | <p>The "app" is the code that you're writing that calls APIs. It can be a web application, a headless server process, a mobile app, or even embedded firmware. Within the AWS developer portal, your app is registered in the My Apps section.</p> <p>Sometimes, "app" refers to your code, other times, it refers to the entity registered in the Dev Portal. It will be clear from context which is meant.</p> <p>When you register your app in the AWS developer portal, you select the APIs that you want to bind to. This ensures that even if you lose control of your authentication credentials, calls cannot be made on behalf to APIs the app was not designed for.</p> <p>A registered app has assigned to it credentials called a "client ID" and "client secret", which are analogous to a username and password, except that you do not select them. These are used to obtain access tokens.</p> | client ID | <p>The "client ID" is essentially your app's username and can be found in the My Apps section. It is an alphanumeric string (e.g., 5zw9v0aUwM7uX3sc0g2DkVvYXt). and it is passed as the value for the client_id query parameter and JGSA attribute.</p> <p>On other platforms, this is sometimes called "consumer key" or "API key"</p> | client secret | <p>The "client secret" is essentially your app's password and can be found next to the client ID in the My Apps section. It is an alphanumeric string (e.g., 756uN3p9vexHf0), and it is passed as the value for the client_secret query parameter.</p> <p>On other platforms, this is sometimes called "consumer secret" or "API secret"</p> | two-legged authentication | <p>This is often considered the default type of authentication, and it is the simplest. When your app needs to call the API without having to access resources that require an end-user's permission. For example, translating a design from one format to another, the communication is directly between your app and AWS. The end user, if there is one, does not need to be aware of the authentication or provide authorization for the app to access any resources.</p> <p>This kind of flow is called "two-legged" because your app and AWS are the two "legs".</p> <p>See the section below about common authentication flows.</p> | three-legged authentication | <p>When an app needs to access resources belonging to an end user, that user must explicitly provide authorization. The typical mechanism, in the case of a web app, is for the app to redirect the user to an Autodesk login page with information about which of the user's resources it wishes to access. The user then provides explicit consent, letting APIs know that the referring app is to be granted the requested access. The user is then redirected back to the app, which can then access the requested resources.</p> <p>This kind of flow is called "three-legged" because your app, AWS, and the end user make up the three "legs" of the flow. Three-legged authentication can be done by using authorization code. When the authorization code grant is used, an Authorization Code is issued to the user to exchange it to obtain Access Token and Refresh Token.</p> <p>See the section below about common authentication flows.</p> | callback URL | <p>In certain contexts, your app needs to hand over part of the authentication flow to AWS. For example, in three-legged authentication for</p> |
| Term   | Meaning  |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| app  | <p>The "app" is the code that you're writing that calls APIs. It can be a web application, a headless server process, a mobile app, or even embedded firmware. Within the AWS developer portal, your app is registered in the My Apps section.</p> <p>Sometimes, "app" refers to your code, other times, it refers to the entity registered in the Dev Portal. It will be clear from context which is meant.</p> <p>When you register your app in the AWS developer portal, you select the APIs that you want to bind to. This ensures that even if you lose control of your authentication credentials, calls cannot be made on behalf to APIs the app was not designed for.</p> <p>A registered app has assigned to it credentials called a "client ID" and "client secret", which are analogous to a username and password, except that you do not select them. These are used to obtain access tokens.</p>   |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| client ID  | <p>The "client ID" is essentially your app's username and can be found in the My Apps section. It is an alphanumeric string (e.g., 5zw9v0aUwM7uX3sc0g2DkVvYXt). and it is passed as the value for the client_id query parameter and JGSA attribute.</p> <p>On other platforms, this is sometimes called "consumer key" or "API key"</p>  |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| client secret  | <p>The "client secret" is essentially your app's password and can be found next to the client ID in the My Apps section. It is an alphanumeric string (e.g., 756uN3p9vexHf0), and it is passed as the value for the client_secret query parameter.</p> <p>On other platforms, this is sometimes called "consumer secret" or "API secret"</p>   |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| two-legged authentication  | <p>This is often considered the default type of authentication, and it is the simplest. When your app needs to call the API without having to access resources that require an end-user's permission. For example, translating a design from one format to another, the communication is directly between your app and AWS. The end user, if there is one, does not need to be aware of the authentication or provide authorization for the app to access any resources.</p> <p>This kind of flow is called "two-legged" because your app and AWS are the two "legs".</p> <p>See the section below about common authentication flows.</p>  |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| three-legged authentication  | <p>When an app needs to access resources belonging to an end user, that user must explicitly provide authorization. The typical mechanism, in the case of a web app, is for the app to redirect the user to an Autodesk login page with information about which of the user's resources it wishes to access. The user then provides explicit consent, letting APIs know that the referring app is to be granted the requested access. The user is then redirected back to the app, which can then access the requested resources.</p> <p>This kind of flow is called "three-legged" because your app, AWS, and the end user make up the three "legs" of the flow. Three-legged authentication can be done by using authorization code. When the authorization code grant is used, an Authorization Code is issued to the user to exchange it to obtain Access Token and Refresh Token.</p> <p>See the section below about common authentication flows.</p>   |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |
| callback URL   | <p>In certain contexts, your app needs to hand over part of the authentication flow to AWS. For example, in three-legged authentication for</p>  |      |         |     |  |           |   |               |  |                           |   |                             |  |              |   |

|                                     |          |   |                        |     |
|-------------------------------------|----------|---|------------------------|-----|
| <input type="checkbox"/>            | 1:16 PM  |  BIM 360 APIs   Autodesk Platform Services   | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 1:16 PM  |  Autodesk Construction Cloud APIs   Autodesk Platform Services                             | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 1:13 PM  |  Autodesk Platform Services  | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 1:12 PM  |  Inventor API   Autodesk Platform Services (APS)   | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 1:11 PM  |  Help   Obtain an access token to use APIs   Autodesk                                      | help.autodesk.com      | ... |
| <input type="checkbox"/>            | 1:07 PM  |  Webhooks API   Autodesk Platform Services (APS)   | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 12:56 PM |  Verify your Autodesk account - mdesanic@gmail.com - Gmail                                 | mail.google.com        | ... |
| <input type="checkbox"/>            | 12:56 PM |  Email Verified  | accounts.autodesk.com  | ... |
| <input type="checkbox"/>            | 12:56 PM |  Inbox (1,821) - mdesanic@gmail.com - Gmail  | mail.google.com        | ... |
| <input type="checkbox"/>            | 12:55 PM |  Autodesk - Create Account   | accounts.autodesk.com  | ... |
| <input type="checkbox"/>            | 12:55 PM |  Sign in   | accounts.autodesk.com  | ... |
| <input type="checkbox"/>            | 12:55 PM |  autodesk developer portal - Google pretraživanje  | www.google.com         | ... |
| <input type="checkbox"/>            | 12:55 PM |  Autodesk Developer Programs   Autodesk Platform Services (APS)                            | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 11:38 AM |  AutoCAD 2025 Developer and ObjectARX Help   The ObjectARX Programming Environment...      | help.autodesk.c...     | ... |
| <input type="checkbox"/>            | 11:37 AM |  AutoCAD 2025 Developer and ObjectARX Help   Overview of ObjectARX   Autodesk              | help.autodesk.com      | ... |
| <input type="checkbox"/>            | 11:37 AM |  AutoCAD 2025 Developer and ObjectARX Help   AutoLISP Developer's Guide (AutoLISP)   A... | help.autodesk.c...     | ... |
| <input type="checkbox"/>            | 11:33 AM |  AutoCAD API   Autodesk Platform Services (APS)  | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 11:32 AM |  AutoCAD 2025 Developer and ObjectARX Help   About AutoCAD JavaScript APIs   Autode...   | help.autodesk.co...    | ... |
| <input type="checkbox"/>            | 11:31 AM |  Reality Capture API   Autodesk Platform Services (APS)                                  | aps.autodesk.com       | ... |
| <input type="checkbox"/>            | 11:07 AM |  available autocad api - Google pretraživanje  | www.google.com         | ... |
| <input type="checkbox"/>            | 11:06 AM |  Autodesk AutoCAD JavaScript API Getting Started Guide: About AutoCAD JavaScri...        | df-prod.autocad360.com | ... |
| <input checked="" type="checkbox"/> | 11:06 AM |  autocad js api documentation - Google pretraživanje                                     | www.google.com         | ... |
| <input type="checkbox"/>            | 11:06 AM |  autocad api documentation - Google pretraživanje  | www.google.com         | ... |
| <input type="checkbox"/>            | 11:05 AM |  AutoCAD 2025 Developer and ObjectARX Help   Acad.TransientManager Function   Auto...    | help.autodesk.co...    | ... |
| <input type="checkbox"/>            | 11:03 AM |  proof of concept - Google pretraživanje   | www.google.com         | ... |
| <input type="checkbox"/>            | 11:03 AM |  Proof of concept - Wikipedia  | en.wikipedia.org       | ... |

Today - Thursday, July 4, 2024

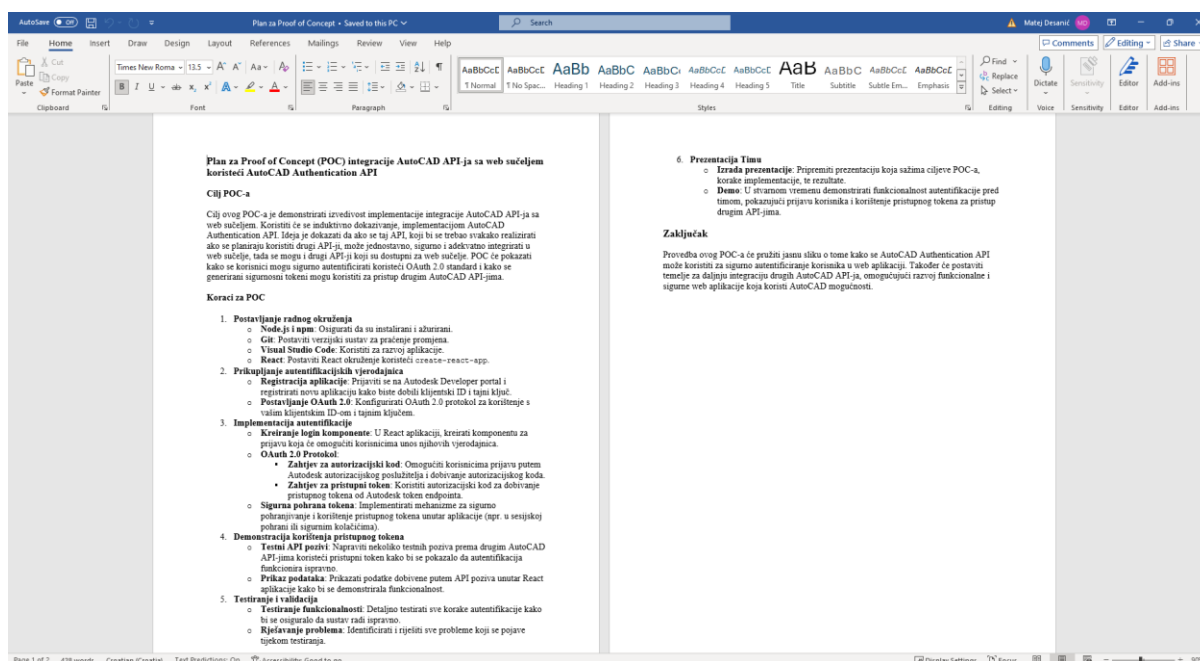
|                          |         |  |                  |   |
|--------------------------|---------|--|------------------|---|
| <input type="checkbox"/> | 2:20 PM |  API Basics   Authentication (OAuth)   Autodesk Platform Services | aps.autodesk.com |  |
|                          |         |  |                  |   |
| <input type="checkbox"/> | 1:54 PM |  Authentication API   Autodesk Platform Services (APS)            | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:46 PM |  Authentication   Autodesk Platform Services                      | aps.autodesk.com |  |
|                          |         |  |                  |   |
| <input type="checkbox"/> | 1:16 PM |  Viewer SDK   Autodesk Platform Services (APS)                    | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Tandem Data API   Autodesk Platform Services (APS)               | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Flow Graph Engine API   Autodesk Platform Services               | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Overview   Data Visualization   Autodesk Platform Services       | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  APIs   Autodesk Platform Services                                | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Data Management API   Autodesk Platform Services (APS)           | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Data Exchange   Autodesk Platform Services (APS)                 | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  BIM 360 APIs   Autodesk Platform Services                        | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:16 PM |  Autodesk Construction Cloud APIs   Autodesk Platform Services    | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:13 PM |  Autodesk Platform Services                                       | aps.autodesk.com |  |
| <input type="checkbox"/> | 1:12 PM |  Inventor API   Autodesk Platform Services (APS)                  | aps.autodesk.com |  |

### 3. Dan – 3.7.2024.

Danas sam došao oko 2 u tvrtku nakon ispita na fakultetu te sam se odmah uhvatio planiranja Proof of concept-a. Kako se dosad nisam susreo s time terminom, morao sam istražiti što je točno Proof of Concept u čemu mi je pomogao mentor. Naučio sam da je Proof of Concept (POC) proces prikupljanja dokaza koji podržavaju izvedivost projekta. Voditelji projekata provode POC u ranim fazama razvoja prije nego što ulože previše vremena i resursa u projekt.

Svrha Proof of Concept-a je demonstrirati održivost projekta proizvodnim timovima, klijentima i drugim dionicima. POC može otkriti nedostatke, što može dovesti do toga da tvrtka revidira ili odustane od projekta. U drugim slučajevima, POC može potvrditi vjerojatnost uspjeha projekta, pružajući dokaze o izvedivosti za daljnji razvoj.

U ovom konkretnom primjeru mentor me uputio na to da demonstriram težinu korištenja AutoCAD API-ja, integracije sa web sučeljem, te da pokušam odrediti vrijednost implementiranja AutoCAD API-ja unutar web sučelja, odnosno je li vrijedno koristiti vrijeme na daljnji razvoj uz pomoć integracije AutoCAD API-ja sa web sučeljem. Odlučeno je da ću na primjeru implementacije API-ja za autentikaciju demonstrirati sve te faktore, jer ako Proof of Concept potvrdi izvedivost i dovoljnu jednostavnost implementacije jednog API-ja koji je odgovoran za funkcioniranje svih ostalih, tada će induktivno implementacija drugih API-ja biti na sličnoj zahtijevnoj razini. Usputno će već biti kreirano rješenje za autentikaciju pa će se moći jednostavno doraditi za implementaciju u daljnjem razvoju.



## 4. Dan – 4.7.2024.

Danas sam se usredotočio na implementaciju osnovnog web sučelja u Reactu, integraciju odabrane AutoCAD biblioteke za autentifikaciju i razvoj jednostavne funkcionalnosti koja demonstrira interakciju između web sučelja i AutoCAD-a.

Počeo sam s implementacijom osnovnog web sučelja u Reactu. Kreirao sam novu React aplikaciju koristeći create-react-app, što je omogućilo brzo postavljanje osnovne strukture projekta. Nakon inicijalnog postavljanja, kreirao sam osnovne komponente potrebne za aplikaciju, kao što su glavna stranica i navigacijska traka. Postavljanje i konfiguracija svih elemenata prošli su bez većih problema, zahvaljujući detaljnom vodiču.

Nakon postavljanja osnovnog sučelja, fokusirao sam se na integraciju odabrane AutoCAD biblioteke. Prvo sam se prijavio na Autodesk Developer portal i registrirao novu aplikaciju kako bih dobio klijentski ID i tajni ključ potrebni za autentifikaciju. Zatim sam konfigurirao OAuth 2.0 protokol u svojoj React aplikaciji kako bih omogućio sigurno generiranje i pohranu sigurnosnih tokena. Ovaj korak bio je ključan za osiguranje sigurnosti aplikacije.

Kao posljednji korak, razvio sam jednostavnu funkcionalnost koja demonstrira interakciju između web sučelja i AutoCAD-a. Kreirao sam login komponentu koja omogućava korisnicima unos vjerodajnica i prijavu putem Autodesk autorizacijskog poslužitelja. Nakon uspješne autentifikacije, aplikacija generira sigurnosni token koji se koristi za testne API pozive prema AutoCAD-u. Prikazao sam podatke dobivene putem API poziva unutar React aplikacije kako bih demonstrirao funkcionalnost i osigurao da autentifikacija radi ispravno. Ovaj proces pokazao je koliko je jednostavno integrirati i koristiti AutoCAD API unutar React aplikacije.

The screenshot shows the Postman interface for configuring an OAuth 2.0 request. At the top, the method is set to 'GET' and the URL is 'https://api.userprofile.autodesk.com/userinfo'. Below this, the 'Authorization' tab is selected, showing the 'Type' as 'OAuth 2.0'. The 'Add authorization data to' dropdown is set to 'Request Headers'. The configuration fields include: 'Client ID' (redacted), 'Client Secret' (input field), 'Code Challenge Method' (set to 'SHA-256'), 'Code Verifier' (set to 'Automatically generated if left blank'), 'Scope' (set to 'viewables:read data:read data:write data:cr...'), 'State' (input field), and 'Client Authentication' (set to 'Send as Basic Auth header'). At the bottom, there are links for 'Clear cookies' and a 'Get New Access Token' button.

```
4 function App() {
10   const handleSubmit = async (e) => {
17     const decoder = new TextDecoder('utf-8');
18     const csv = decoder.decode(result.value);
19     console.log("CSV Data:", csv);
20     const parsedData = Papa.parse(csv, { header: true }).data;
21     console.log("Parsed Data:", parsedData);
22
23
24     // Check if entered credentials match any entry in the CSV
25     const user = parsedData.find(
26       (row) => row.username === username && row.password === password
27     );
28
29     if (user) {
30       setMessage('Login successful!');
31     } else {
32       setMessage('Invalid username or password.');

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR




webpack compiled successfully


```

## Authorize application



poc-async is requesting permission for the following information:

 mdesanic

**Data**  
View your data  
Manage your data  
Write data  
Search across your data

**Storage**  
Create new buckets  
View your buckets  
Update your buckets  
Delete your buckets

**Others**  
View your viewables

Allow

---

## Tutorials

Get step-by-step guidance to learn how you can complete the most common API workflows.

### Get a client ID and secret

Learn how to register an application and obtain credentials to call APS APIs.

---

[➔ View tutorial](#)

### Get a 2-legged token

Use your client ID and secret to verify the access between 2 applications.

---

[➔ View tutorial](#)

### Get an ID token

Learn to get an ID JSON Web Token with a 60-minute lifespan to verify user claims through trusted token signatures.

---

[➔ View tutorial](#)

### Get a 3-legged token with authorization code grant (PKCE)

Learn to use the authorization code grant type with PKCE to allow a web app to act on a user's behalf, obtaining both access and refresh tokens.

---

[➔ View tutorial](#)

## Code samples

Explore practical examples that demonstrate specific API functionality implementation and build upon the existing code to customize it according to your needs.

### Get a 3-legged authorization code grant

Use the authorization code grant type to obtain both access and refresh tokens for your web app to act on a user's behalf.

---

[➔ View code](#)

### Validate an access JSON Web Token

This sample shows how to obtain and parse the JWT, verifying the signature used to sign the token as well as the claims added to it.

---

[➔ View code](#)

### Use 3-legged OAuth v.2 with Postman

Learn how to set up and use 3-legged OAuth 2.0 authentication in Postman for APS applications.

---

[➔ Read blog post](#)



## 5. Dan – 5.7.2024.

Danas sam se fokusirao na finalizaciju Proof of Concepta (PoC) i pripremu prezentacije za tim.

Prvo sam završio sve preostale korake u izradi PoC-a. To je uključivalo finalne testove funkcionalnosti integracije AutoCAD Authentication API-ja s web sučeljem u Reactu. Osigurao sam da svi elementi aplikacije rade ispravno, uključujući prijavu korisnika i sigurno generiranje te pohranu sigurnosnih tokena. Napravio sam nekoliko testnih API poziva kako bih demonstrirao uspješnu autentifikaciju i prikazao podatke unutar aplikacije. Tijekom testiranja riješio sam nekoliko manjih problema s konfiguracijom, što je omogućilo nesmetano funkcioniranje svih komponenti.

Nakon dovršetka PoC-a, posvetio sam se pripremi prezentacije. Uključio sam pregled istraženih mogućnosti integracije, demonstraciju razvijenog PoC-a, zaključke o izvedivosti i potencijalnim izazovima te preporuke za daljnji razvoj. Priprema prezentacije obuhvatila je organiziranje svih bitnih informacija u logičan i koherentan slijed, kako bi tim mogao lako pratiti i razumjeti proces te rezultate mog rada.

Na kraju dana, održao sam prezentaciju rezultata pred timom. Pokazao sam ključne aspekte istraženih mogućnosti integracije AutoCAD API-ja, demonstrirao razvijeni PoC, te objasnio zaključke o izvedivosti i potencijalnim izazovima. Predstavio sam i preporuke za daljnji razvoj, sugerirajući korake koje bi tim trebao poduzeti kako bi se integracija dodatnih API-ja nastavila glatko i efikasno. Tim je bio zadovoljan rezultatima, a prezentacija je pružila jasan smjer za buduće aktivnosti.

