POSTMAN GALAXY

VeriToll

Using Cookies to Store Your Postman Secrets

Lightning Talk on February 4, 2021

Miguel A. Calles

www.veritoll.com

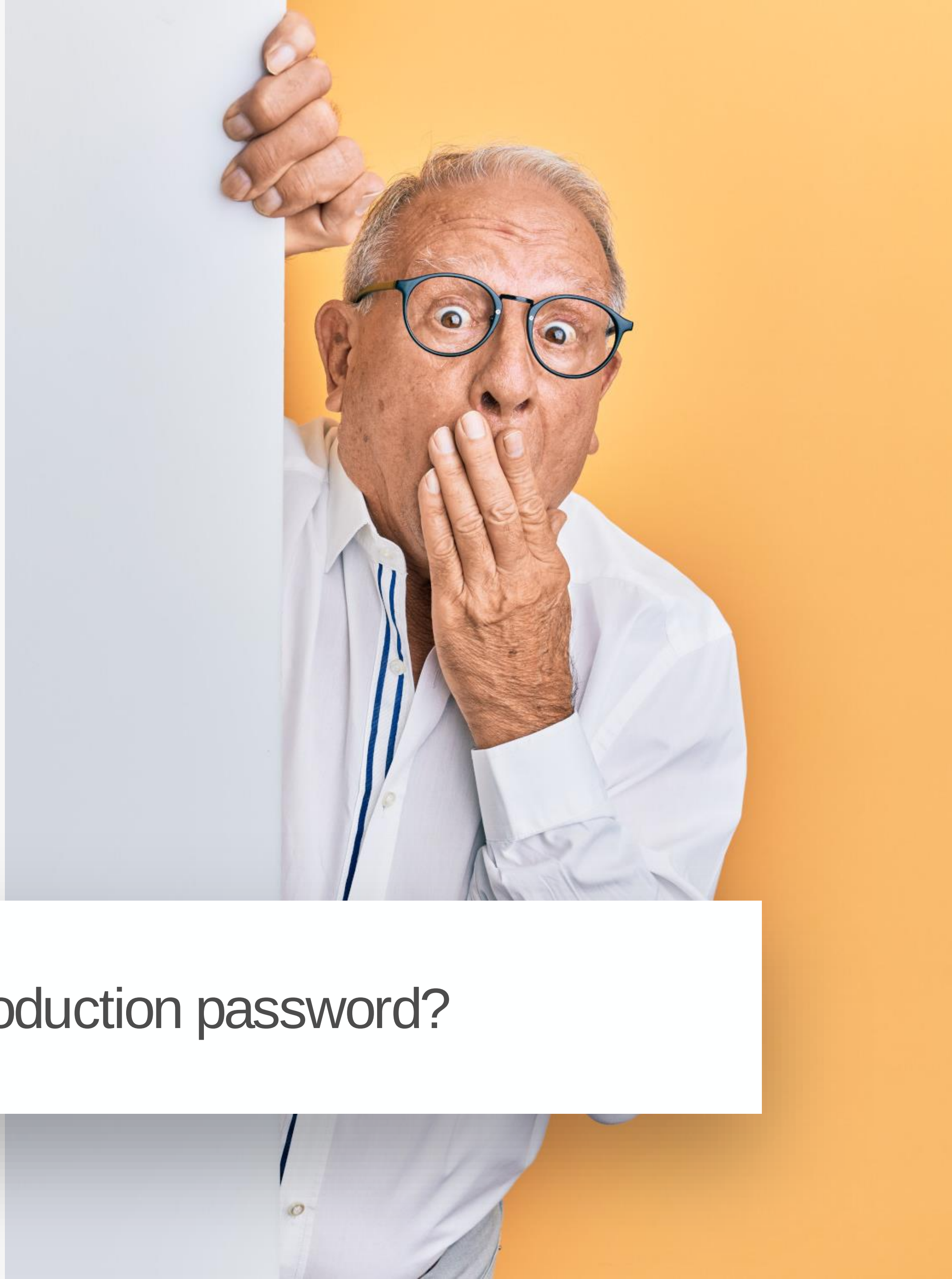
Let's suppose

We are testing the production environment.

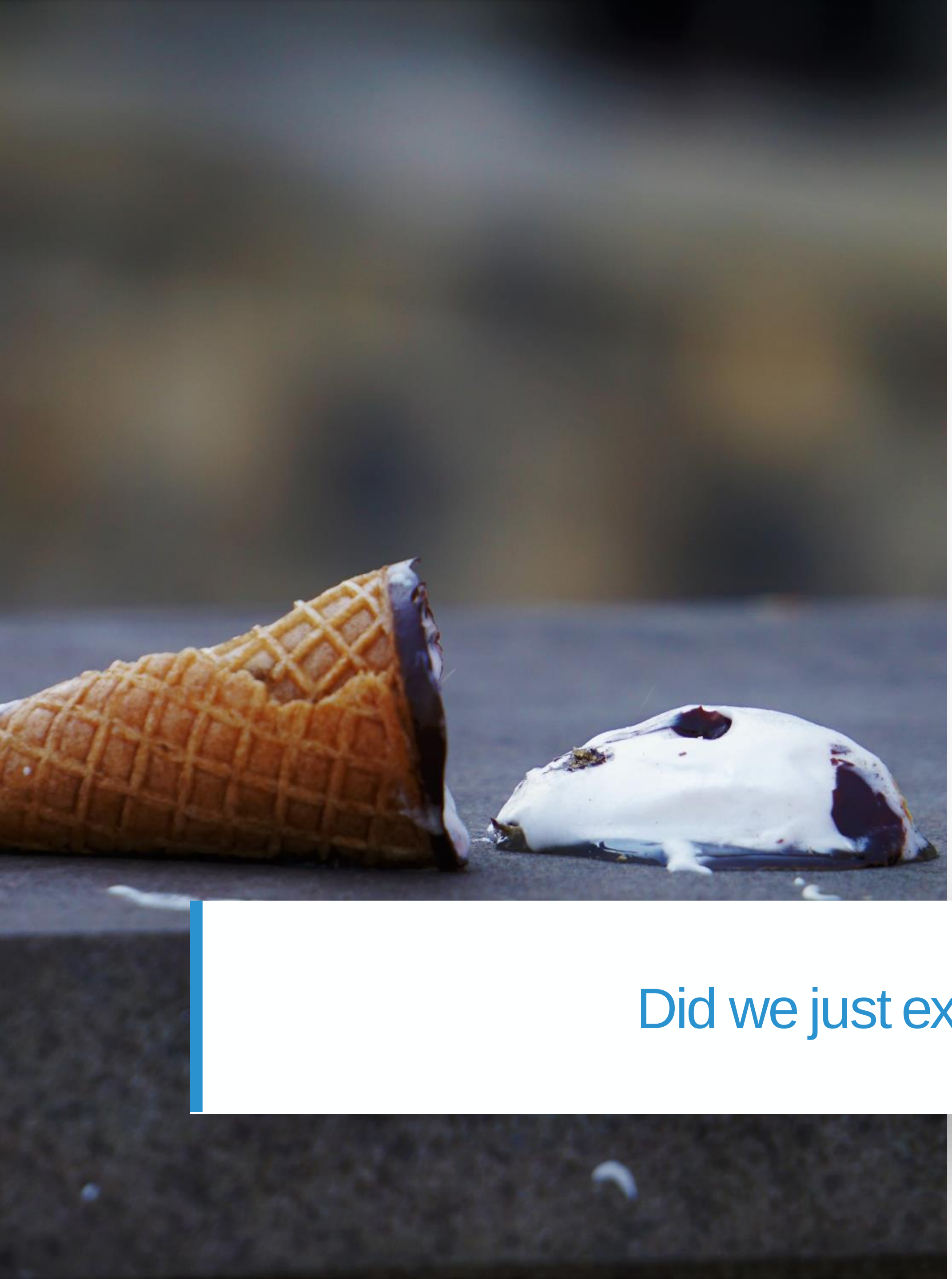
We want to test an API in a team collection.

We enter our actual username and password.

We complete the test and move on.



Did we just forget to remove our production password?



Let's suppose

We make a collection public.

We decide to make a "quick" change.

We "temporarily" add an API key.

We get a phone call from our boss.

Did we just expose an API key to the public?

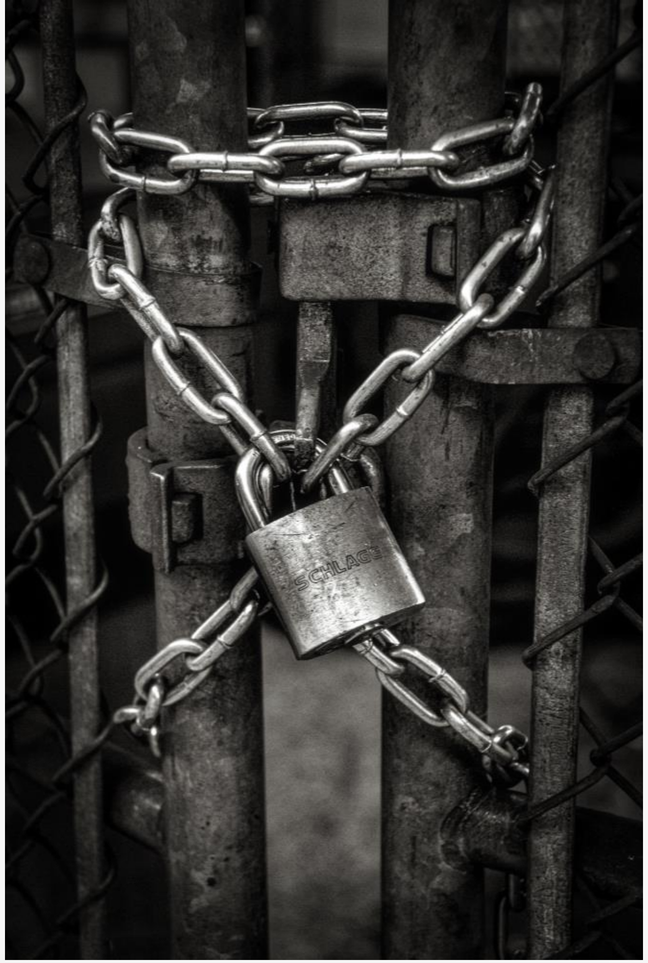
Postman's Security

Strong encryption at-rest and in-transit

Postman Sessions

Role-based access control (RBAC)

Strong security program



Learn more at https://www.postman.com/security/

Shared Responsibility

Protect your account, installation, and computer

Implement user roles with RBAC

Use Postman Sessions

Be careful what you share

Learn more at https://www.postman.com/security/shared-responsiblity/

Sessions + Cookies

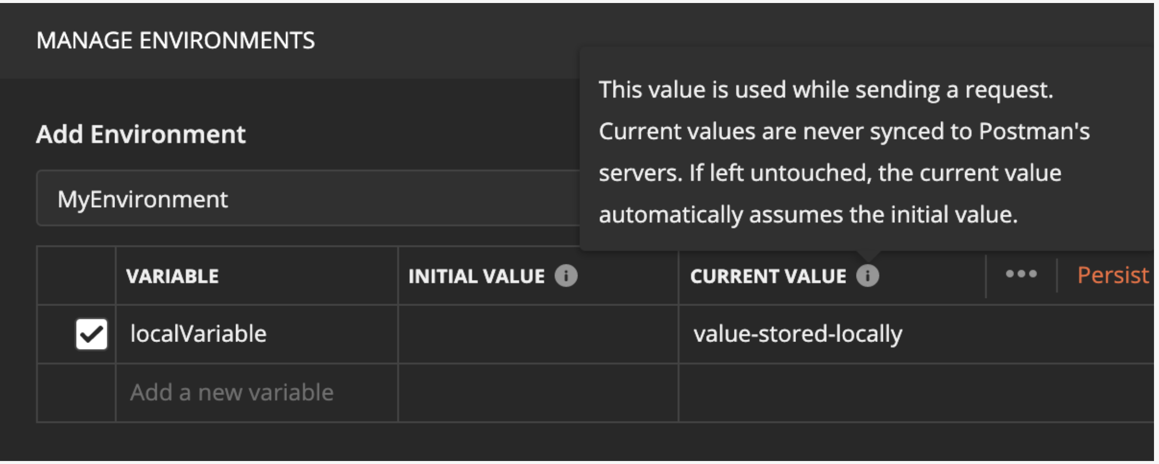
Environments (obviously) have no encryption in-use

Avoid syncing with Sessions

Use cookies as a local data store

Store secrets in cookies and use them with sessions

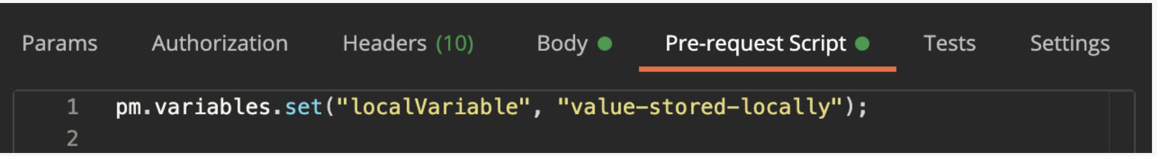
Using Sessions



MyEnvironment

This value is used while sending a request. Current values are never synced to Postman's servers. If left untouched, the current value automatically assumes the initial value.

	VARIABLE	INITIAL VALUE	CURRENT VALUE	••• Persist
☑	localVariable		value-stored-locally	
	Add a new variable			



```
pm.variables.set("localVariable", "value-stored-locally");
```

Done with the UI and scripts.

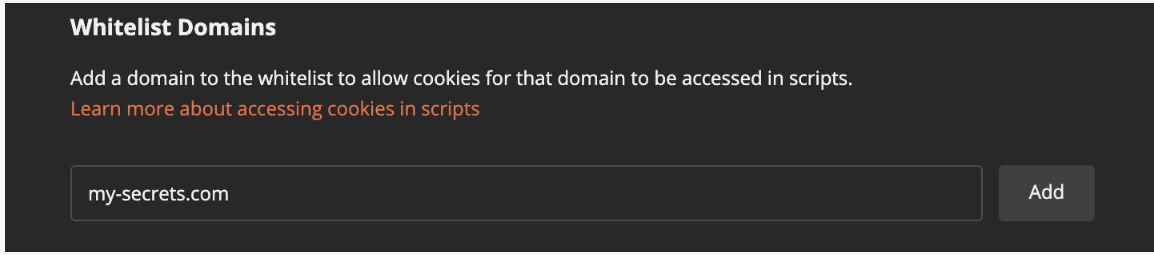
Must carefully avoid setting Initial Value

Learn more at https://learning.postman.com/docs/sending-requests/variables/

Using Cookies

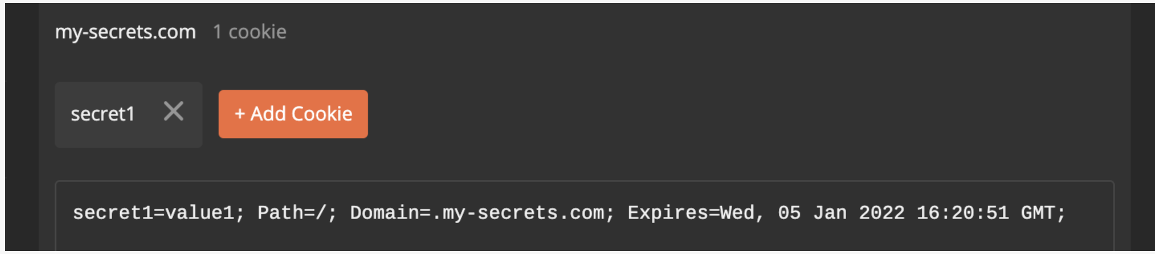
Done with the UI and scripts

Must whitelist domain



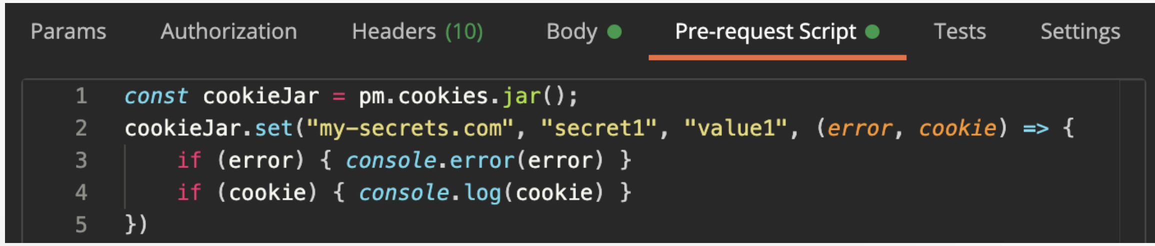
Whitelist Domains

Add a domain to the whitelist to allow cookies for that domain to be accessed in scripts.
Learn more about accessing cookies in scripts



my-secrets.com 1 cookie

```
secret1=value1; Path=/; Domain=.my-secrets.com; Expires=Wed, 05 Jan 2022 16:20:51 GMT;
```

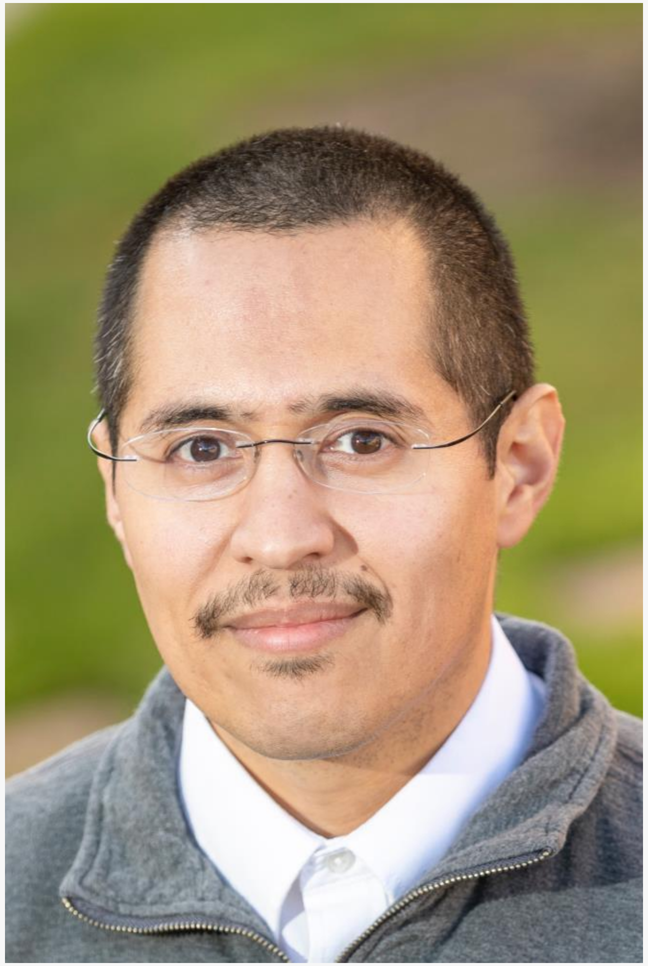


```javascript
const cookieJar = pm.cookies.jar();
cookieJar.set("my-secrets.com", "secret1", "value1", (error, cookie) => {
    if (error) { console.error(error) }
    if (cookie) { console.log(cookie) }
})
```

Learn more at https://learning.postman.com/docs/sending-requests/cookies/

Demo

Putting it all together



About me

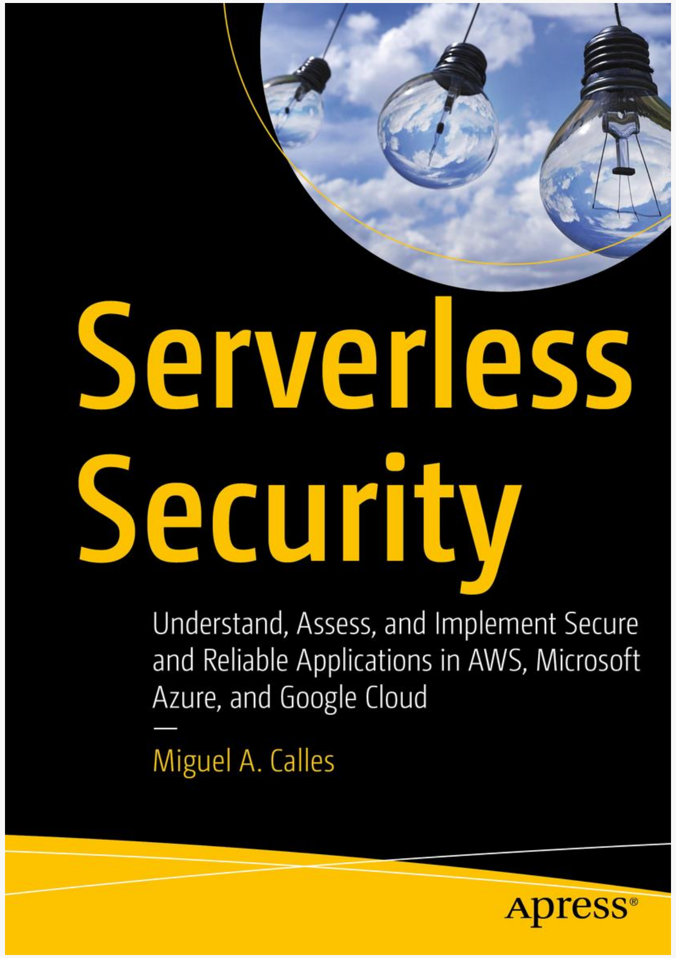
Principal Solutions and Security Engineer

Published Author

https://MiguelACallesMBA.com

https://ServerlessSecurityBook.com

https://www.linkedin.com/in/miguel-a-calles-mba/



Serverless Security

Understand, Assess, and Implement Secure and Reliable Applications in AWS, Microsoft Azure, and Google Cloud

—

Miguel A. Calles

Apress®

VeriToll is Tolling-as-a-Service

VeriToll

The major benefits of Software-as-a-Service coupled with our innovative technology in order to create a new service model for the tolling industry. We provide the people, process, and technology to deliver integrated services within the tolling & transportation industry.

MOBILE, CLOUD, CROWDSOURCED

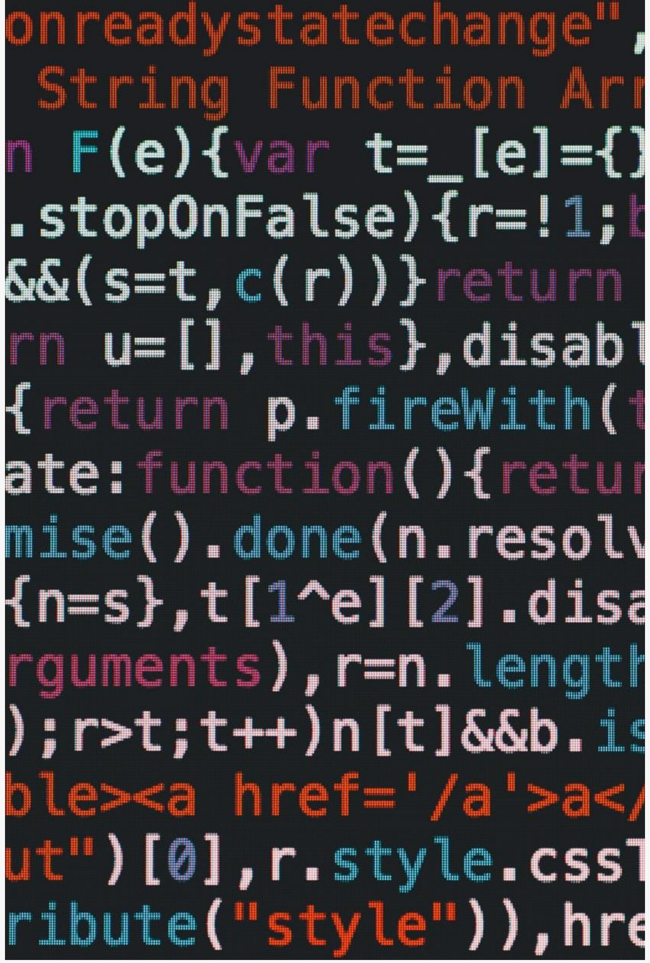
AuditToll

Continuous and automatic end-to-end testing of tolling systems utilizing crowdsourced data from drivers' mobile phones & GPS. Helping toll operators mitigate revenue loss.

CrowdToll

Human quality image review and quality assurance services for tolling operations by crowdsourcing the image review services and utilizing patented AI/ML techniques.

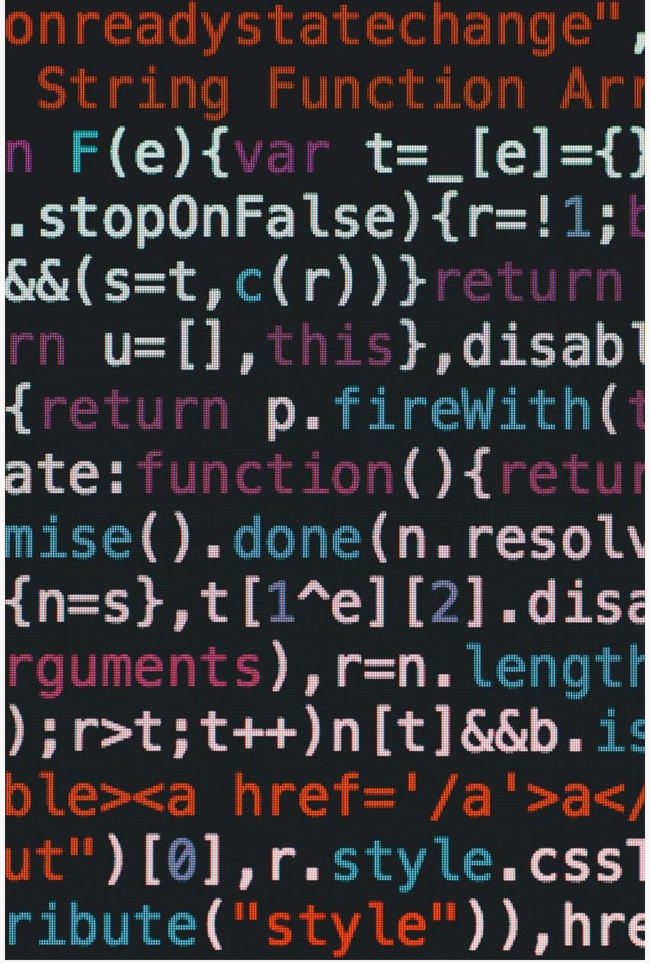
The power of innovation. The power of people.



```
onreadystatechange",
 String Function Arr
n F(e){var t=_[e]={}
.stopOnFalse){r=!1;b
&&(s=t,c(r))}return
rn u=[],this},disabl
{return p.fireWith(t
ate:function(){retur
mise().done(n.resolv
{n=s},t[1^e][2].disa
rguments),r=n.length
);r>t;t++)n[t]&&b.is
ble><a href='/a'>a</
ut")[0],r.style.cssT
ribute("style")),hre
```

Code · Pre-request Script · Using Cookie

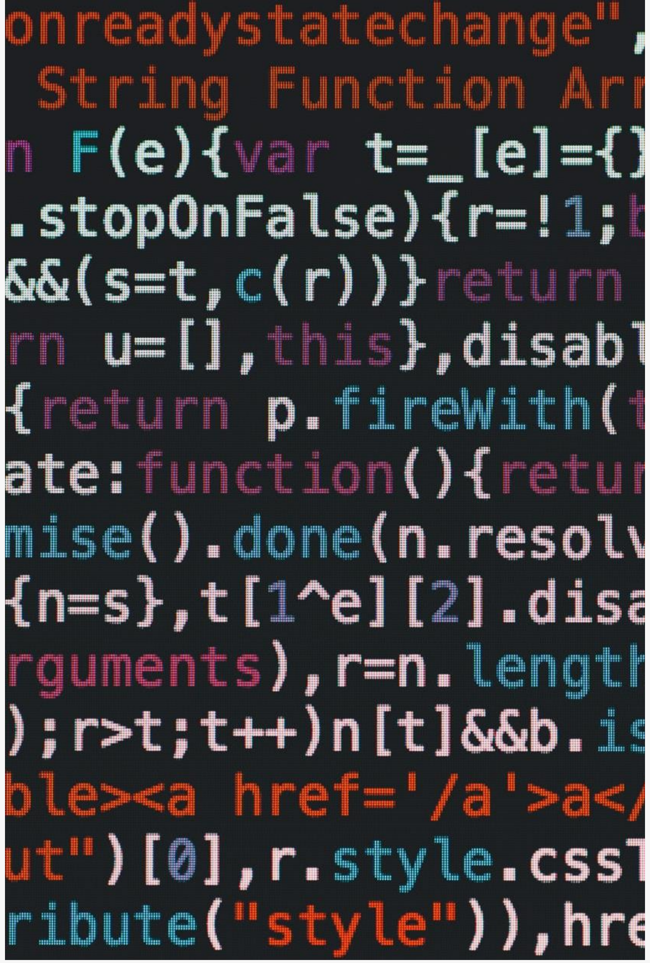
```
const cookieJar = pm.cookies.jar();
const cookieName = "xApiKey"
const domain = "postman.galaxy.demo"
cookieJar.get(domain, cookieName, (error, cookie)
=> {
  if (error) {
    console.error(error);
    pm.variables.set(cookieName, "error");
  }
  if (cookie) {
    pm.variables.set(cookieName, cookie);
  } else {
    console.error("Cookie is missing")
    pm.variables.set(cookieName, "missing");
  }
});
```



```
onreadystatechange",
 String Function Arr
n F(e){var t=_[e]={}
.stopOnFalse){r=!1;b
&&(s=t,c(r))}return
rn u=[],this},disabl
{return p.fireWith(t
ate:function(){retur
mise().done(n.resolv
{n=s},t[1^e][2].disa
rguments),r=n.length
);r>t;t++)n[t]&&b.is
ble><a href='/a'>a</
ut")[0],r.style.cssT
ribute("style")),hre
```

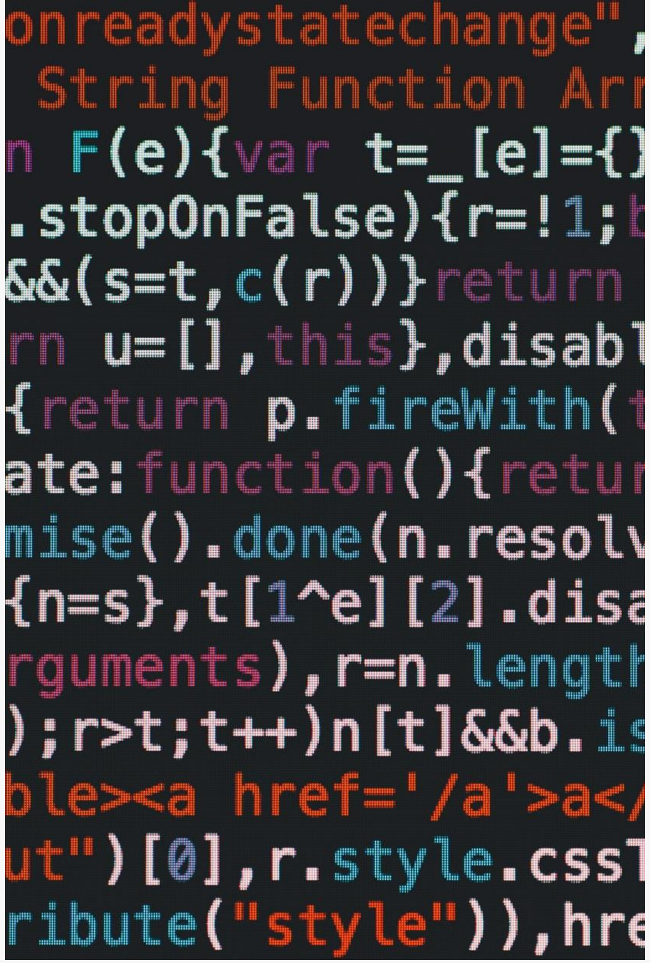
Code · Tests Script · Clearing Session Variable

```
pm.variables.unset("xApiKey");
```



Code · Pre-request Script · CryptoJS · Part 1

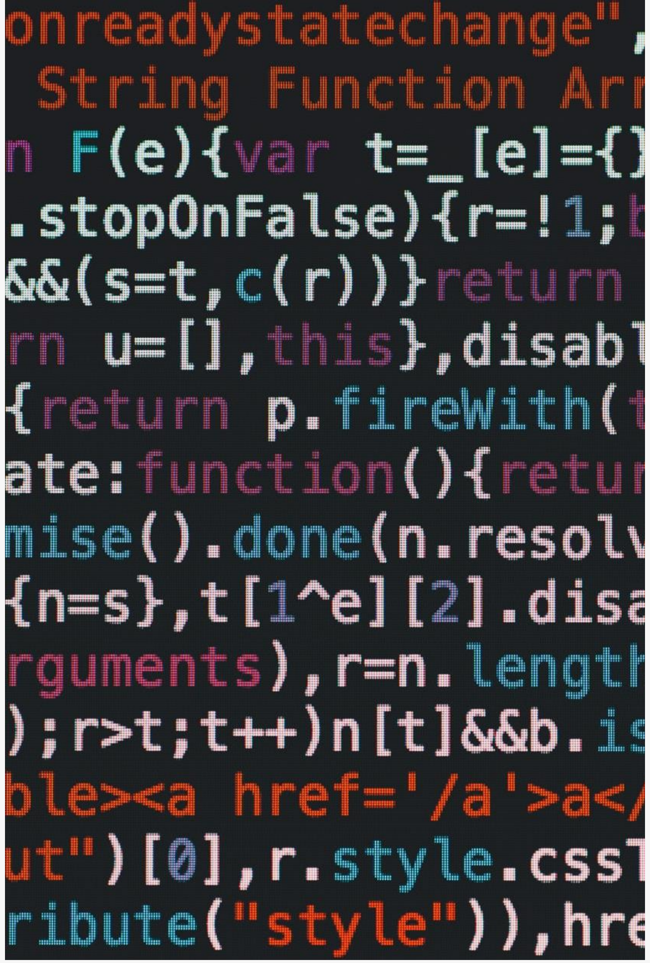
```
// https://postman-quick-reference-guide.readthedocs.io/en/latest/libraries.html
const cookieJar = pm.cookies.jar();
const sessionVarName = "xApiKey";
const cookieName = "secretKey";
const domain = "postman.galaxy.demo";
```



```
onreadystatechange",
 String Function Arr
n F(e){var t=_[e]={}
.stopOnFalse){r=!1;b
&&(s=t,c(r))}return
rn u=[],this},disabl
{return p.fireWith(t
ate:function(){retur
mise().done(n.resolv
{n=s},t[1^e][2].disa
rguments),r=n.length
);r>t;t++)n[t]&&b.is
ble><a href='/a'>a</
ut")[0],r.style.cssT
ribute("style")),hre
```

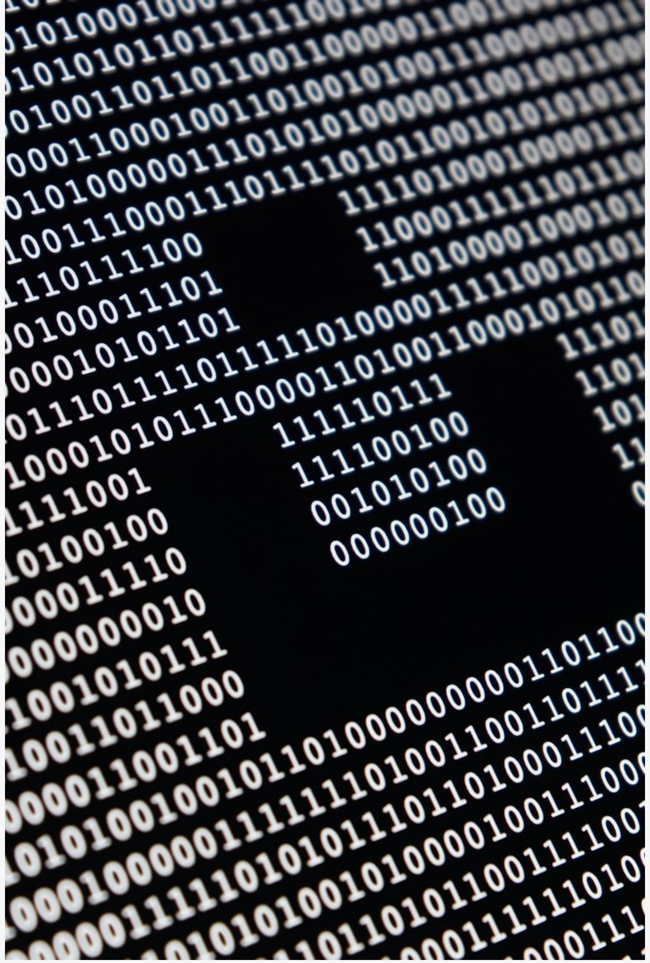
Code · Pre-request Script · CryptoJS · Part 2

```
cookieJar.get(domain, cookieName, (error,
secretKey) => {
  if (error) {
    console.error(error);
    pm.variables.set(sessionVarName, "error");
  }
  if (secretKey) {
    // encryption
    const encryptedText =
CryptoJS.AES.encrypt('<data-to-encrypt>',
secretKey).toString();
    console.log('encryptedText', encryptedText);
```



Code · Pre-request Script · CryptoJS · Part 3

```
    // decryption
    console.log('secretKey', secretKey);
    const xApiKeyEnc = pm.environment.get('x-api-
key-enc');
    console.log('xApiKeyEnc', xApiKeyEnc);
    const xApiKey =
CryptoJS.AES.decrypt(xApiKeyEnc,
secretKey).toString(CryptoJS.enc.Utf8);
    console.log('xApiKey', xApiKey);
    pm.variables.set(sessionVarName, xApiKey);
  } else {
    console.error("Cookie is missing")
    pm.variables.set(sessionVarName, "missing");
  }
});
```



Credits

Photo by krakenimages on Unsplash

Photo by Sarah Kilian on Unsplash

Photo by John Salvino on Unsplash

Photo by Erika Fletcher on Unsplash

Photo by Alexander Sinn on Unsplash

Photo by Christina Branco on Unsplash

Photo by Scott Sanker on Unsplash

Photo by Markus Spiske on Unsplash