



# Build a Simple REST API Web App From Scratch

Mark Bykerk Kauffman, Senior Software Engineer, Blackboard Inc.

# Forward-Looking Statement

Statements regarding our product development initiatives, including new products and future product upgrades, updates or enhancements represent our current intentions, but may be modified, delayed or abandoned without prior notice and there is no assurance that such offering, upgrades, updates or functionality will become available unless and until they have been made generally available to our customers.



## About Me

---

- Blackboard's Partner Software Engineer/  
5yrs+
- Former Learn Admin & Dev @ CSU Chico
- Scientist/Engineer @ Hewlett-Packard
  - Report Generation System and Method,  
US Patent# 7031979
- Grandfather, 6X marathons, hiking &  
gaming

# Agenda

## Django Web Application

- Security
- Session

## BbRest

- Developer Portal Registration
- Integrate With Blackboard Learn

## Django and BbRest

- Make REST API calls to Blackboard Learn
- 2-Legged OAuth & 3-Legged OAuth



# Demo!

# Agenda

## Web Application

- Securtiy

## BbRest

- Developer Portal Registration
- Integrate With Blackboard Learn

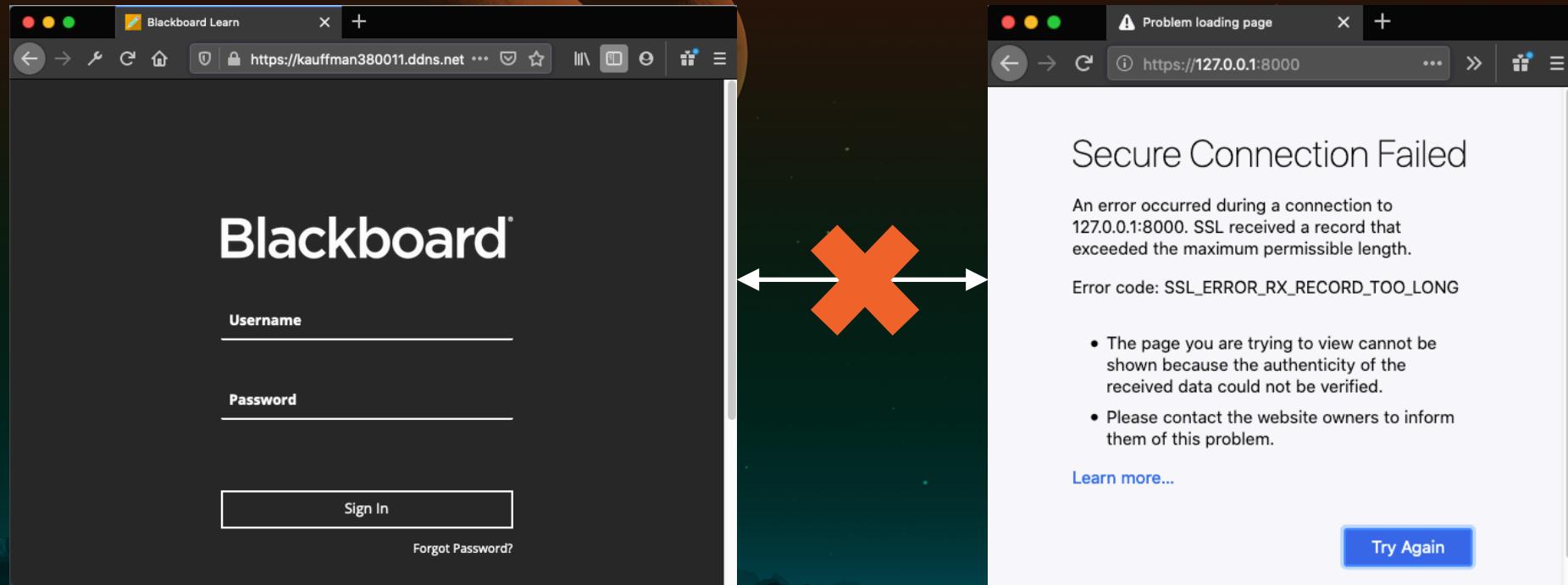
## Django and BbRest

- Make REST API calls to Blackboard Learn
- 2-Legged OAuth & 3-Legged Oauth

# Secure Django Applications

Search For: “how to get ssl cert and secure django web application”

# SSL/TLS



# SSL/TLS - ngrok Reverse Proxy to Django on localhost:8000

---

- Registered a DNS Name On Dreamhost: avinynet.com
  - Got SSL Cert For www.avinyet.com
  - Created a CNAME Record For [www.avinyet.com](http://www.avinyet.com) to ngrok Account
- ngrok Account
  - Custom Domain Reservation, CNAME Target @ Dreamhost DNS
  - \$ ~/ngrok tls -region=us -hostname=www.avinyet.com -key ~/avinet.rsaprivatekey.pem -crt ~/avinet.fullchaincert.pem 8000
  - ngrok “secure introspectable tunnel” to Django on localhost

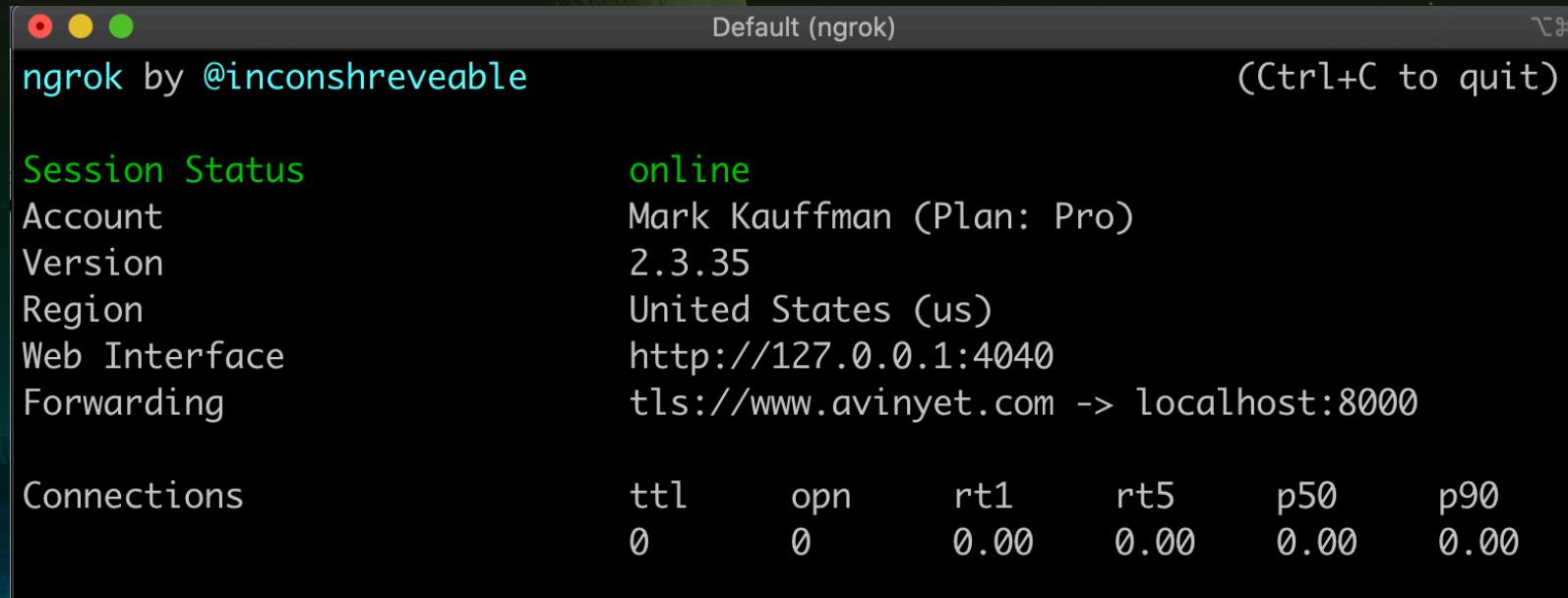
# SSL Required

The screenshot shows a web browser window with the URL <https://dashboard.ngrok.com/endpoints/domains>. The page title is "Reserved Domains". A subtext explains that reserved subdomains are secure and whitelabelable. A search bar and a "Reserve a Domain" button are visible. A table lists two reserved domains:

ID	Region	Domain	Description	Target
rd...2vov0w	US	apotelesm.ngrok.io	localhost	
rd...33ahwt	US	www.avinyet.com	cname on dreamhost where we got the cert	CNAME

# SSL/TLS Required

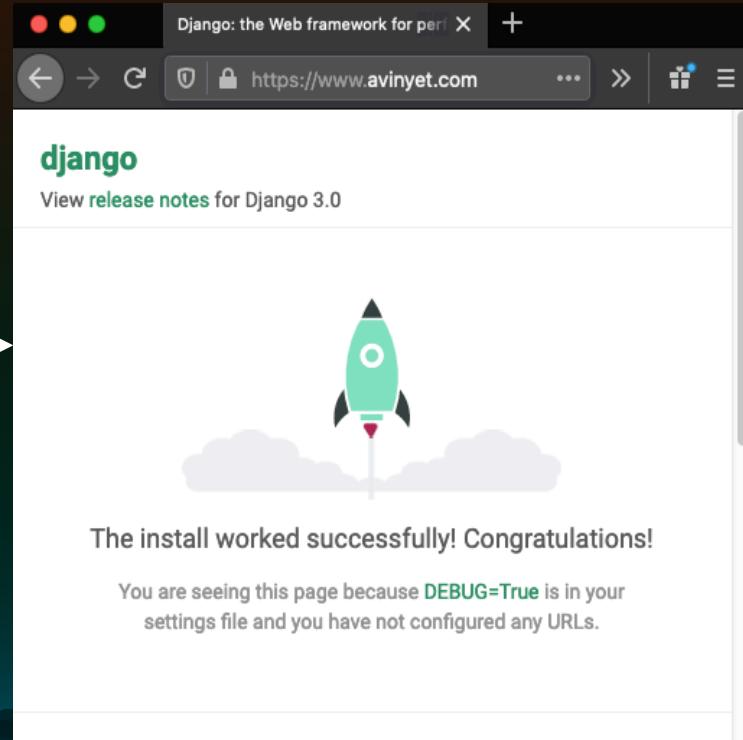
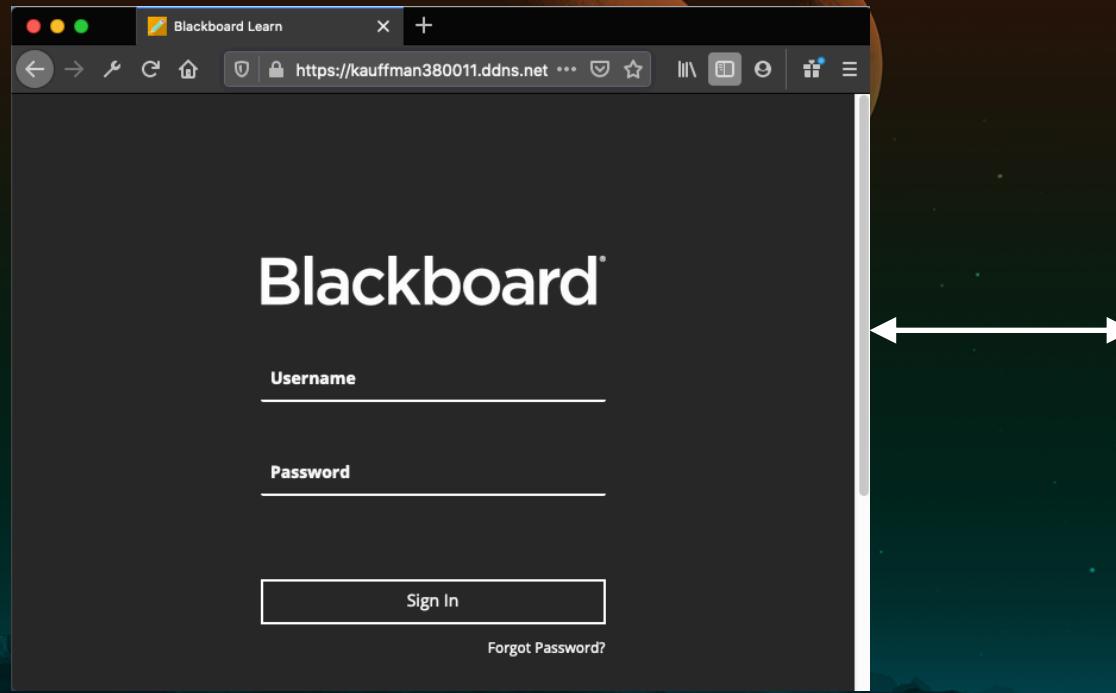
```
$ ~/ngrok tls -region=us -hostname=www.avinyet.com -key  
~/avinet.rsaprivatekey.pem -crt ~/avinet.fullchaincert.pem 8000
```



The screenshot shows the ngrok command-line interface running in a terminal window. The title bar reads "Default (ngrok)". The window content includes:

- ngrok by @inconschreveable
- (Ctrl+C to quit)
- Session Status: online
- Account: Mark Kauffman (Plan: Pro)
- Version: 2.3.35
- Region: United States (us)
- Web Interface: http://127.0.0.1:4040
- Forwarding: tls://www.avinyet.com -> localhost:8000
- Connections: ttl open rt1 rt5 p50 p90  
0 0 0.00 0.00 0.00 0.00

# SSL/TLS



Django settings.py: ALLOWED\_HOSTS = ['127.0.0.1', 'localhost', 'www.avinyet.com']



# Django

- “Reassuringly secure.”
  - Django takes security seriously and helps developers avoid many common security mistakes
- Session Management
  - Can Save Python Objects on Session
- Settings.py

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

# REST Applications for Blackboard Learn

Register and Setup & BbRest

# Register on developer.blackboard.com

The screenshot shows a web browser window for developer.blackboard.com. The address bar displays the URL. The top navigation bar includes links for Apps, Bookmarks, and various saved items like 'Bb REST Entitlements', 'BbBdWork', 'Docker', and 'DevCon2020'. Below the navigation is a secondary menu with links for Blackboard, Community, Explore APIs, My Apps, and My Groups. On the right, there are 'My Account' and 'Logout' buttons. The main content area features a large heading 'Manage Your Applications'. Below it, there are two tabs: 'My Applications' (which is selected and highlighted in purple) and 'My Site Registrations'. A callout text 'Click (+) to Register a New Application' points to a blue plus sign button. A descriptive paragraph explains the functionality of managing applications. The background of the page has a space-themed graphic with stars and a planet.

developer.blackboard.com/portal/applications

Blackboard Community Explore APIs My Apps My Groups My Account Logout

# Manage Your Applications

Text

My Applications My Site Registrations

Click (+) to Register a New Application

You may edit an application's Name and Description. You can view the Schools using your application. You may also request your Application Key. Finally, you may also delete your application.

- No LTI Needed For This Workshop
- Domains: localhost[,<any domains your app may run on>]

# Register on developer.blackboard.com

The screenshot shows a web browser window for developer.blackboard.com. The URL in the address bar is developer.blackboard.com/portal/applications/create. The page title is "Blackboard". The navigation menu includes "Community", "Explore APIs", "My Apps", and "My Groups". On the right, there are links for "My Account" and "Logout". The main content area displays the creation of a new application named "ABbRestApp". It shows a section titled "Important note!" with the text: "The secret and tool private key are only shown once. Make note of the application key, secret and tool private key and store them in a safe and secure location." Below this, three fields are shown: "Application key" (18849c74-7c2a-464a-94e7-db2a024ffeb6), "Secret" (MdU3hf9sBzdmW05QXxQwKrX179QwlbdE), and "Application ID" (3d8575d7-0b89-4d84-882b-88cbc64b8849). Each field has a "Copy" button to its right.

- PAUSE!!
- Save This Information

# Setup on Learn

Administrator Panel

REST API Integrations

Create Integration

## Create Integration

\* Indicates a required field.

### GENERAL INFORMATION

\* Application ID

appId from registration

\* Learn User

user/sysrole => entitled

Browse...

\* End User Access

Yes  No

\* Authorized To Act As User

Yes  No  Service Default (No)

- <https://github.com/mdeakyne/BbRest>
  - Automagically Makes All Blackboard Learn REST API Endpoints Available for Your Use
  - Gets Access Token via Two-legged Authentication or Three-legged Authentication
  - Caches & Refreshes Access Token When Necessary
  - Obviates the Need for Writing Requests to /learn/api/public/v.... etc.
  - Under Active Development - Use Pull Requests

```
git clone https://github.com/mark-b-kauffman/weblearn.git
cd weblearn/
python -m venv env
source env/bin/activate
pip install -r requirements.txt
pip install --upgrade pip
python
>>> from bbrest import BbRest
>>> key = 'your-app-key'
>>> secret = 'your-app-secret'
>>> fqdn = 'your learn server fqdn'
>>> learnfqdn = "kauffman380011.ddns.net"
>>> bb = BbRest(key, secret, f"https://{learnfqdn}")
>>> resp = bb.GetUser('userName:mkauffman')
>>> resp
<Response [200]>
>>> resp.json()
{'id': '_5_1', 'uuid': 'b0de25aa4db04043a79c667a5b0adbb3', 'externalId': 'mkauffman', 'dataSourceId': '_2_1...
```

# BbRest & Django

The Best of the Best

[https://www.youtube.com/watch?v=\\_huL5ynal8Y](https://www.youtube.com/watch?v=_huL5ynal8Y)

- <https://github.com/mark-b-kauffman/weblearn>
- settings.py
  - Installed Apps: twoleg, threeleg
  - ALLOWED\_HOSTS
- urls.py - Routing
- views.py - Logic
- templates (<html>) - Render

- Show Me the Code!

# Questions?

# DEVCON 20