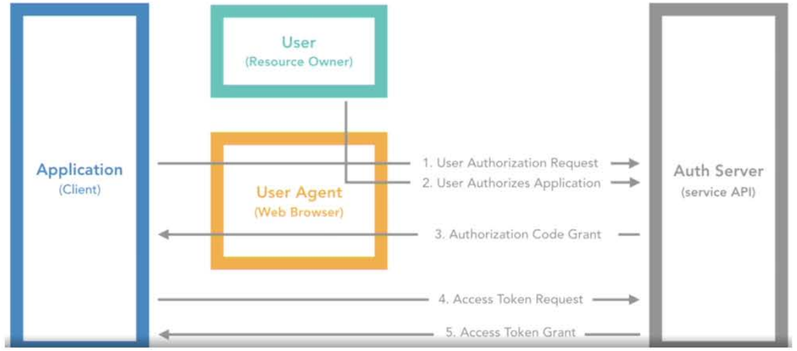
What about legacy applications?

Resource Owner Password Flow

This is not the flow you're looking for.

Authorization Code Flow



The Basics

- The user never saw the access token
- The application had the user's credentials
- Do we trust the application and its developers?

This defeats the purpose of OAuth.

There is one valid use case

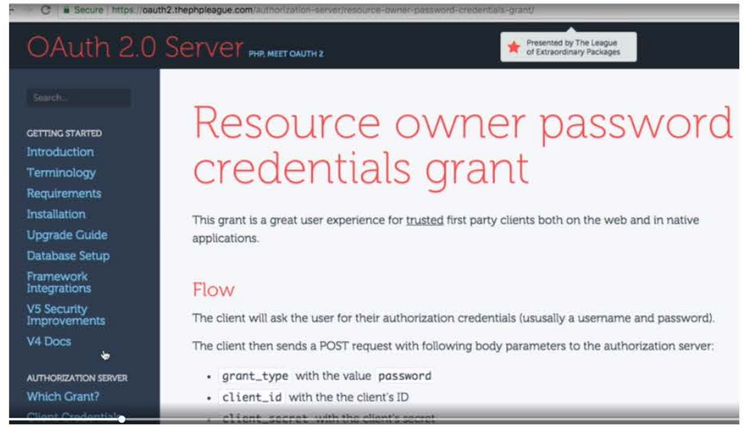
Wrapping legacy systems in an OAuth interface to bring them into the same security practices

This should be a *temporary* state.

New applications should never use Resource Owner Password Flow.

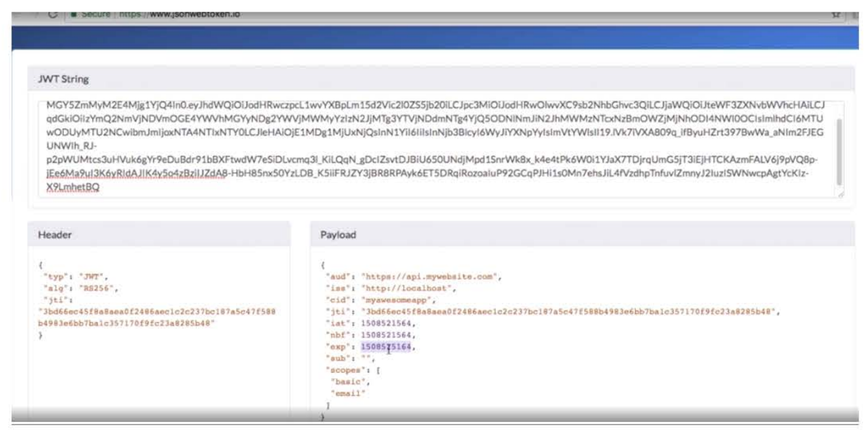
Time for some code

(that hopefully you never need).




```
~
curl -X POST "http://localhost:4444/password.php/access_token" -d
curl: option -d: requires parameter
curl: try 'curl --help' or 'curl --manual' for more information
~
curl -X POST "http://localhost:4444/password.php/access_token" -d "grant_type=password&client_id=m
yawesomeapp&client_secret=abc123&username=keith&password=keithisawesome&scope=basic%20email"
{"token_type":"Bearer","expires_in":3600,"access_token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6I
jgxZjdiNjhmYWI3ZGEyMjU0NGFlZGMzNzJkNTM1Y2Q3YjhkNWM1ZTZjMjQ1N2FjNGUzNDE5MTRkODliYjE5OTkyZmRiNTViNWU5N2Z
mYTY0In0.eyJhdWQiOiJodHRwczpcL1wvYXBpLm15d2Vic2l0ZS5jb20iLCJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3QiLCJjaWQiO
iJteWF3ZXNvbWVhcHAiLCJqdGkiOiI4MWY3YjY4ZmFiN2RhMjI1NDRhZWRjMzcyZDUzNWNkN2I4ZDVjNWU2YzI0NTdhYzRlMzQxOTE
0ZDg5YmIxOTk5MmZkYjU1YjVlOTdmZmE2NCIsImlhdCI6MTUwODUyMTU1NCwibmJmIjoxNTA4NTIxNTU0LCJleHAiOjE1MDg1MjUxN
TQsInN1YiI6IiIsInNjb3BlcyI6WyJiYXNpYyIsImVtYWlsIl19.L86ussEfD5KavX5Esbk4lJecXnQTb9OpXk5GBfRLi50kAnnCAm
dCSYiVGpC6b-gbAWNUwt8DXENJSj-PRzJvtSr7PBU9j8PwJ3hJpkBSyxk1xjUBL3agGkMHC5NAFwNQolv51UiwoBA9kIWGtWNnvTw0
3yB4q5D88ztLURkuNiTi_yg7m8hmbkgC6-pYLJSpWGfEP9oJOAuLFM9AfVLY9-TaNItLvRaoE54JuNYB1u7NjruX8MsAOCdMQ7Vs7v
3L7AyUVNJyRoVtfb6araaD9sGwaRmUTYd3EaJ9guYh80h73MSOlLW7n1XP32SrhOQPHGMGHjUSjotF5xdHFBwWIA"}~
curl -X POST "http://localhost:4444/password.php/access_token" -d "grant_type=password&client_id=m
yawesomeapp&client_secret=abc123&username=keith&password=keithisawesome&scope=basic%20email" | ~/jq
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1028  100   906  100   122   4620    622 --:--:-- --:--:-- --:--:--  4646
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6IjNiZDY2ZWM0NWY4YThhZWEwZjI0ODZhZWMxYzJ
jMjM3YmMxODdhNWM0N2Y1ODhiNDk4M2U2YmI3YmExYzM1NzE3MGY5ZmMyM2E4Mjg1YjQ4In0.eyJhdWQiOiJodHRwczpcL1wvYXBpL
m15d2Vic2l0ZS5jb20iLCJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3QiLCJjaWQiOiJteWF3ZXNvbWVhcHAiLCJqdGkiOiIzYmQ2NmV
jNDVmOGE4YWVhMGYyNDg2YWVjMWMyYzIzN2JjMTg3YTVjNDdmNTg4YjQ5ODNlNmJiN2JhMWMzNTcxNzBmOWZjMjNhODI4NWI0OCIsI
mlhdCI6MTUwODUyMTU2NCwibmJmIjoxNTA4NTIxNTY0LCJleHAiOjE1MDg1MjUxNjQsInN1YiI6IiIsInNjb3BlcyI6WyJiYXNpYyI
sImVtYWlsIl19.lVk7iVXA809q_ifByuHZrt397BwWa_aNIm2FJEGUNWIh_RJ-p2pWUMtcs3uHVuk6gYr9eDuBdr91bBXFtwdW7e5i
DLvcmq3l_KiLQqN_gDcIZsvtDJBiU650UNdjMpd1SnrWk8x_k4e4tPk6W0i1YJaX7TDjrqUmG5jT3iEjHTCKAzmFALV6j9pVQ8p-jE
e6Ma9uI3K6yRIdAJIK4y5o4zBziIJZdA8-HbH85nx50YzLDB_K5iiFRJZY3jBR8RPAyk6ET5DRqiRozoaIuP92GCqPJHi1s0Mn7ehs
JiL4fVzdhpTnfuvlZmnyJ2IuzISWNwcpAgtYcKIz-X9LmhetBQ"
}
```



Resource Owner Password Flow Security Considerations

Security Considerations

- Protect the client_secret (backend apps only)
- Must use secure communications like TLS
- Protect the access token
- All resource servers must validate the access tokens

Security Considerations

- The actual risk is much bigger

  Your user is giving their credentials to the application.

  Do we want to train users to put their credentials anywhere?

  Do we trust the application and all the developers?

- Scopes don't protect us here
- Revoking access doesn't work either

There is one valid use case.

Wrapping legacy systems in an OAuth interface to bring them into the same security practices

!

If you can, avoid the Resource Owner Password Flow.