

Web Payments Team

PUBLIC

Introducing DGAPI

TPAC October 2021 #wpwg

rouslan@google.com (he/him)

How we got here ⁰¹

Development ⁰²

UX ⁰³

Next steps ⁰⁴

Who what now

Why DGAPI?

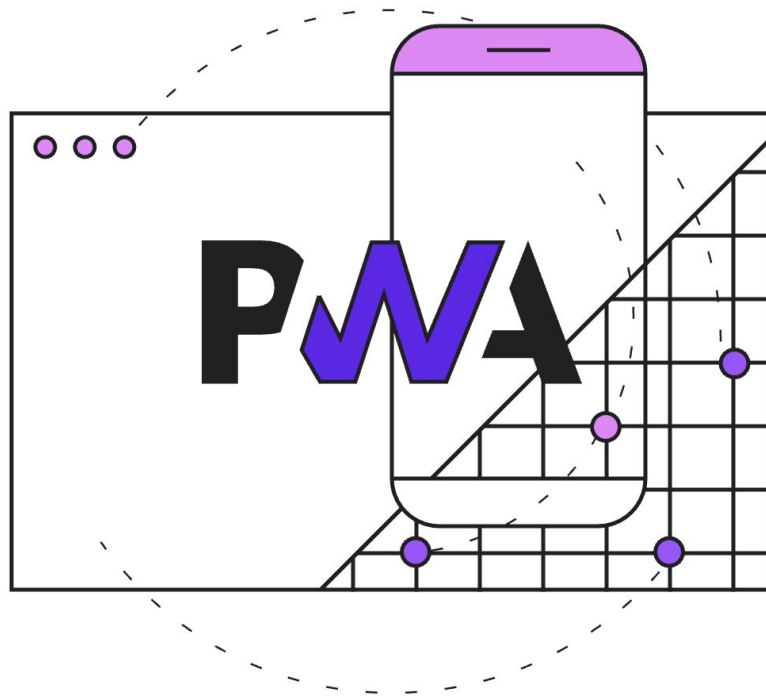
Backstory

Trusted Web Application =
Progressive web app +
Android wrapper.

TWAs can be listed in Android
app stores.

Some app stores provide
seamless purchase flow.

DGAPI enables access to that.



Seamless purchase flow: No need to enter CC details or re-authenticate the user.

Some stores require own purchase flows for DG.
No DGAPI => Difficult to build web apps for app stores.

Example stores: Play, Samsung, Microsoft, Apple...

<https://web.dev/progressive-web-apps/>

Play Billing is the first implementer of DGAPI.

How we got here ⁰¹

Development ⁰²

UX ⁰³

Next steps ⁰⁴

Developers, developers, developers

github.com/GoogleChromeLabs/bubblewrap

What's the developer experience?

Dev ex

Build PWA.

Bubblewrap it.

Publish it to the app store.

Add digital goods to the app store's developer console.

Add DGAPI calls to the PWA. 

Add non-DGAPI payments for the WWW.

```
// Simplified code with the main concepts.  
// More accurate code is in the appendix.
```

```
// Rejects outside valid service context.  
await window.getDigitalGoodsService(  
  'https://service-id.example');
```

```
// Build the shop page.  
await service.getDetails(  
  ['shiny_sword', 'gem']);
```

```
// Invoke the purchase flow.  
new PaymentRequest([ {  
  supportedMethods:  
    'https://service-id.example',  
  data: {sku: 'gem'}} ])  
.show();
```


How we got here ⁰¹

Development ⁰²

UX ⁰³

Next steps ⁰⁴

Wow

What's the user experience?

w3.org

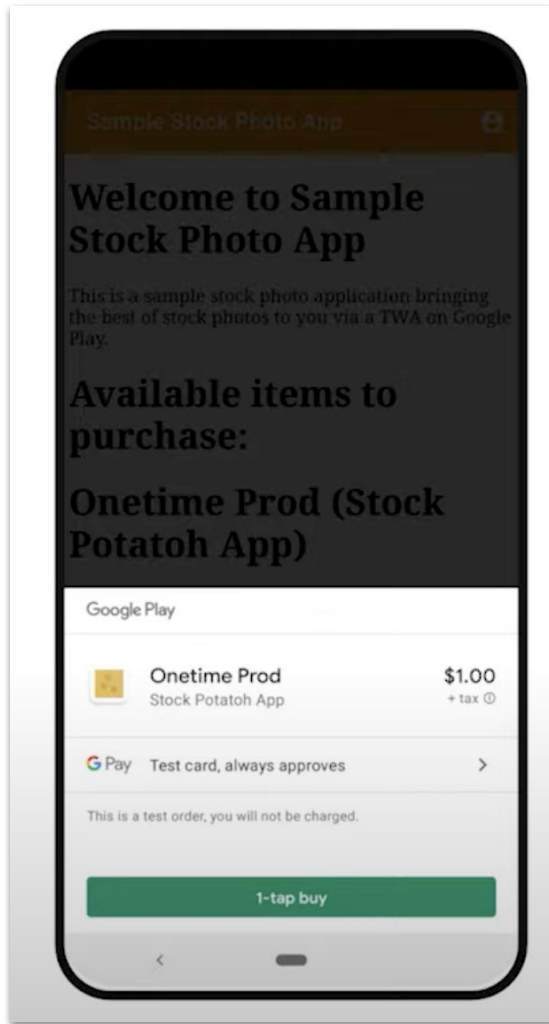
You ex

Find an app in a store.

Install it.

Purchase something.

(User is agnostic to the app being a PWA.)



How we got here ⁰¹

Development ⁰²

UX ⁰³

Next steps ⁰⁴

What is the future of DGAPI?

Next steps

Collect feedback from the origin trial.

Publish draft spec in WICG.

Iterate with more developer feedback.

Collaborate with other user agents and app stores.

Ship it.

<https://www.w3.org/community/wicg/>



Call for participants:

User agents

App stores

Apps

Thank you

Appendix 1: Render product details

```
if (window.getDigitalGoodsService === undefined) {
  return;
}

try {
  const service =
    await window.getDigitalGoodsService('https://play.google.com/billing');

  if (service === null) {
    return;
  }

  const skuDetails = await itemService.getDetails(['shiny_sword', 'gem']);
  for (item of skuDetails) {
    // Format the price according to the user locale.
    const localizedPrice = new Intl.NumberFormat(
      navigator.language,
      {style: 'currency', currency: item.price.currency}
    ).format(item.price.value);

    renderProductDetails(
      item.itemId, item.title, localizedPrice, item.description);
  }
} catch (error) {
  return;
}
```

Appendix 2: Make a purchase

```
async function makePurchase(service, sku) {
  const paymentMethods = [{
    supportedMethods: 'https://play.google.com/billing',
    data: {
      sku: sku,
    }
  }];

  try {
    const request = new PaymentRequest(paymentMethods);
    const paymentResponse = await request.show();
    const {purchaseToken} = paymentResponse.details;

    if (await validateAndAcknowledgePurchaseOnBackend(purchaseToken)) {
      await paymentResponse.complete('success');
    } else {
      await paymentResponse.complete('fail');
    }
  } catch(e) {
    // The purchase failed, and we can handle the failure here. AbortError
    // usually means a user cancellation
  }
}
```