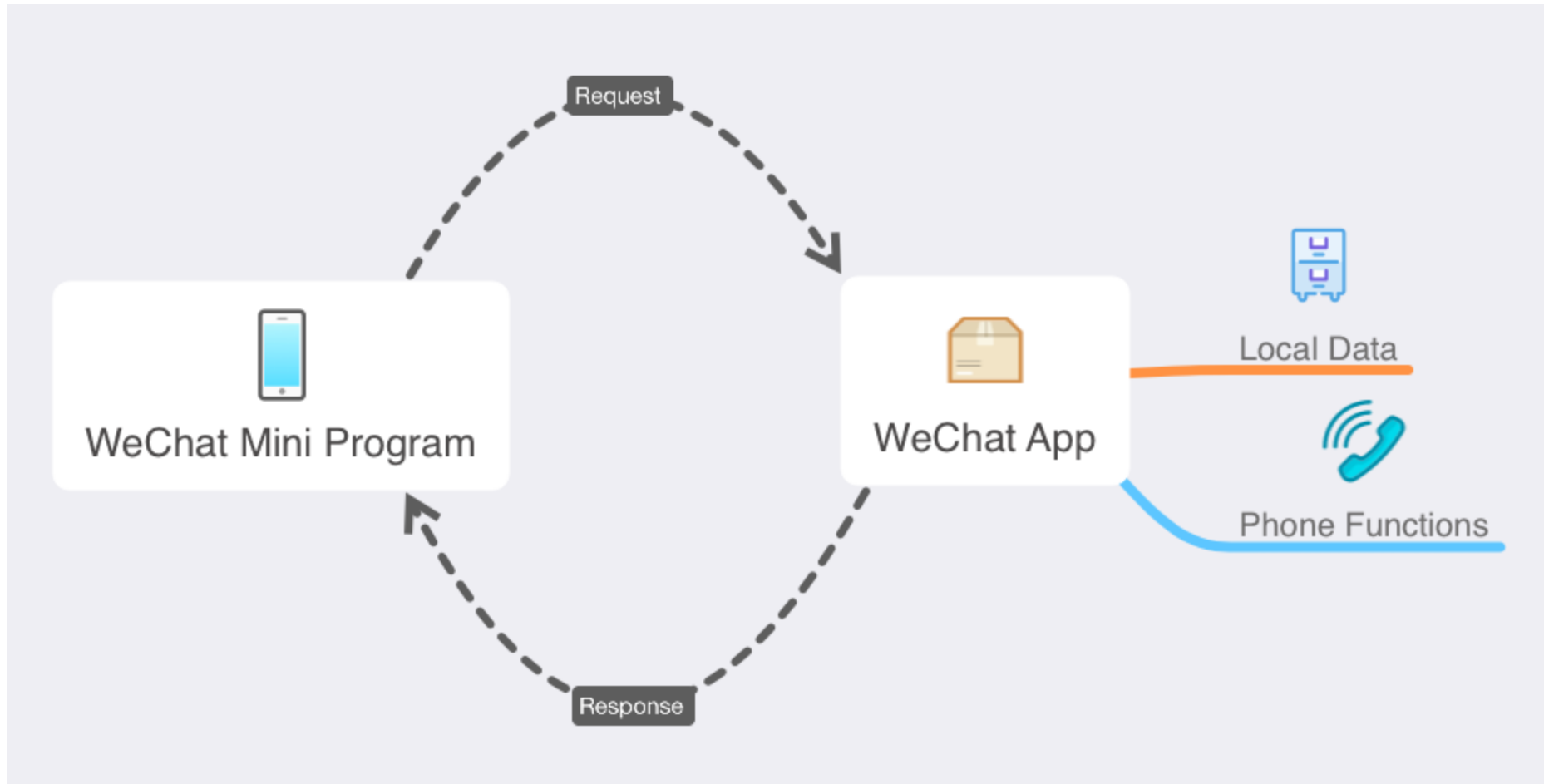


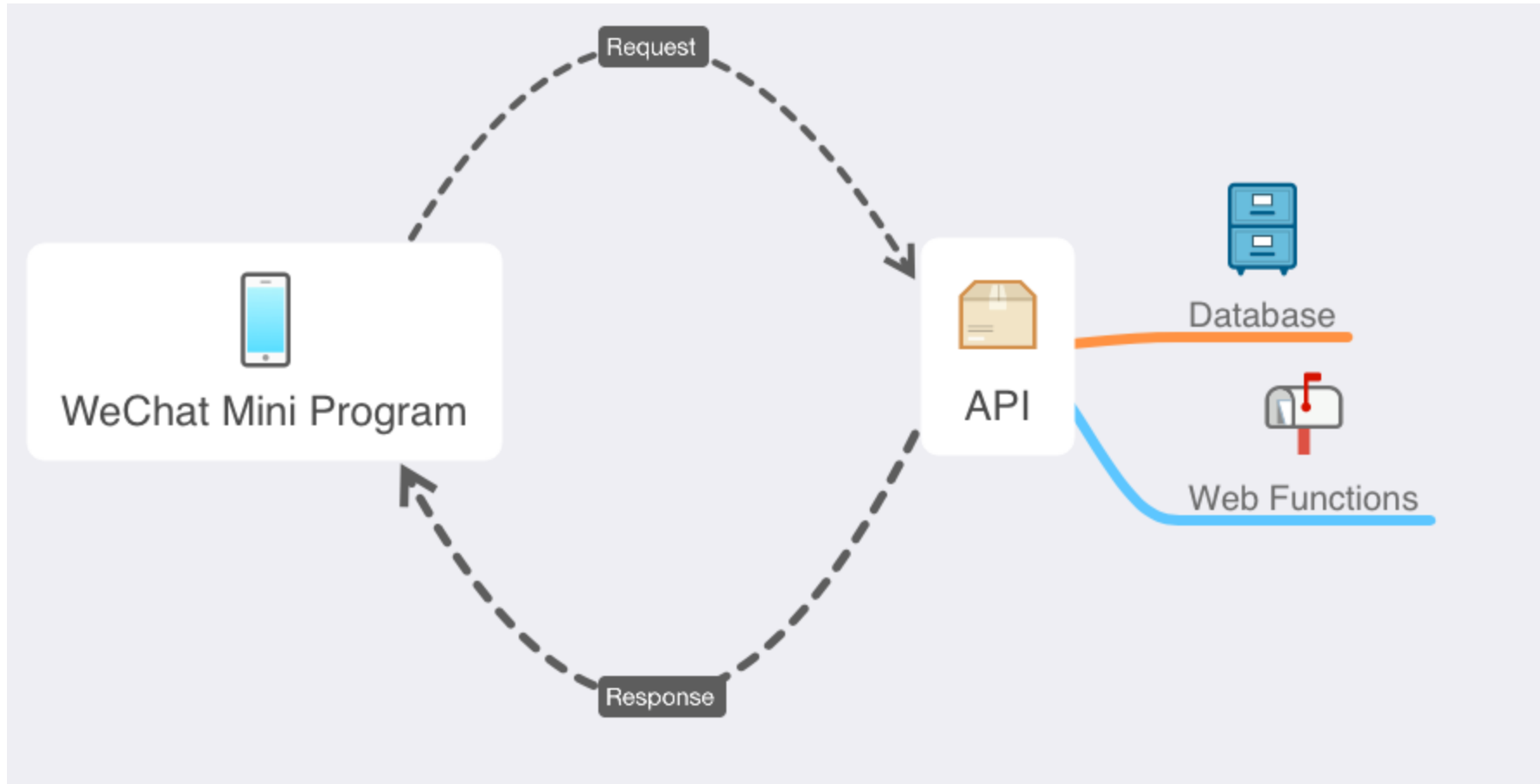
# Mini Programs 小程序 with API

API WITH DATA FOR WECHAT MPS

# OUR FRONTEND SO FAR



# OUR FRONTEND WITH API TODAY



## APIs ALLOW YOU TO

- Provide data for client (e.g. an app: web, native, Wechat Mini Program)
- Provide service to customers (e.g. sms, payment): later course

**NO BIG DEAL**

It's just a new set of **Endpoints...**

...which return **JSON** format for contents of cells

**THEN JSON => Mini Program**

Mini Program makes **API Requests...**

...and uses **JSON** from API instead of static data (e.g globalData in app.js)



# JSON

```
{  
  "header": {  
    "title": "The JSON example",  
    "descriptionText": "This is some title text."  
  },  
  "content": {  
    "title": "The content example text",  
    "elements": [  
      {  
        "title": "The first element",  
        "mainText": "First element main text",  
        "additionalText": "First element additional text"  
      },  
      {  
        "title": "The second element",  
        "mainText": "Second element main text",  
        "additionalText": "Second element additional text"  
      }  
    ]  
  }  
}
```

# REST-FUL API

Purpose	Verb	URI Pattern	Table#Action
all stories	GET	/stories	stories#index
create story	POST	/stories	stories#create
one story	GET	/stories/:id	stories#show
edit story	PUT	/stories/:id	stories#update
delete story	DELETE	/stories/:id	stories#destroy

4 Verbs: GET, POST, PUT, DELETE

## 6 STEPS OF USING API

1. Use API key
2. Specify endpoint
3. Attach request data
4. Send request and **wait** for response
5. Receive data from response
6. Handle the data

## **INDEX : 1ST ENDPOINT**

For showing all the stories

## **1. Use API token (or key)**

From API provider (when creating account)

Something like:

```
API_KEY: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
TOKEN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Use like:

```
header: {'Authorization': 'Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'}
```

or as params or form data

```
data: {'api_key': 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxx'}
```

For today's exercise - no API token needed

## 2. Specify endpoint

Restful: Verb and Path

```
GET /api/v1/stories
```

Combine with host: `https://fml.shanghainwogeng.com`

Get endpoint: `https://fml.shanghainwogeng.com/api/v1/stories`

```
// /pages/index/index.js
```

```
Page({
```

```
  //...
```

```
  onLoad: function (options) {
```

```
    // Save reference to page
```

```
    let page = this;
```

```
    //...
```

```
    const request = {
```

```
      url: `https://fml.shanghainwogeng.com/api/v1/stories`,
```

```
      method: 'GET' // If no method, default is GET
```

```
    }
```

```
  }
```

```
  //...
```



### **3. Attach request data**

Data can be sent with request as a json object

No need for our index page (we want everything).

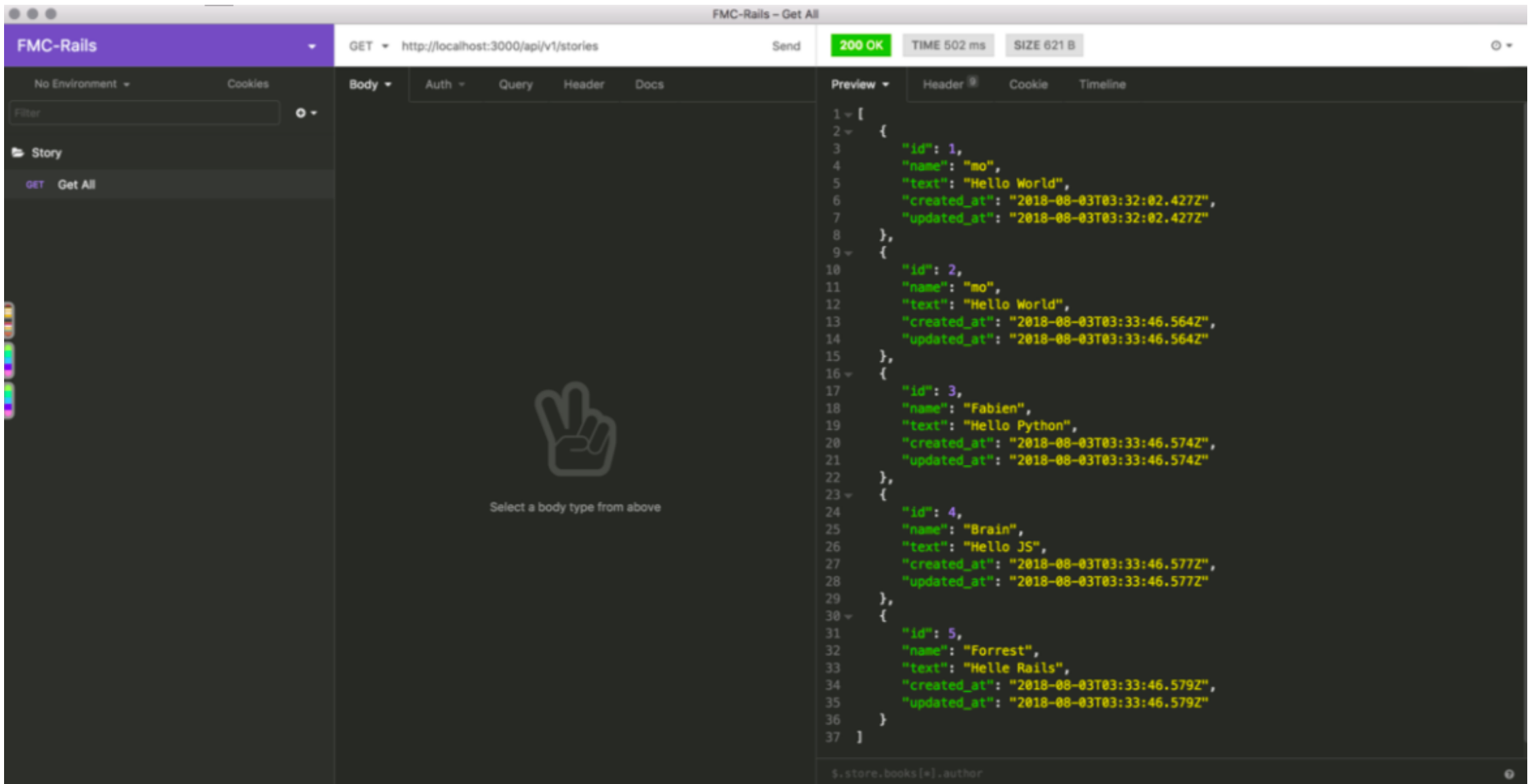
Possible for filtering in the future:

```
// /pages/index/index.js
// in onLoad
// ...
let filter = {
  include: 'My name',
}

const request = {
  url: `https://fml.shanghaiwogeng.com/api/v1/stories`,
  method: 'GET',
  data: filter // Not today, but later in the course
}
```

For index, we don't have any data

4. Send request and **wait** for response



Tools: You can use [Postman](#) or [Insomnia](#)

In browser: `https://fm1.shanghainwogeng.com/api/v1/stories`

In WeChat MP:

```
// /pages/index/index.js

Page({
  //...
  onLoad: function (options) {
    // Save reference to page
    let page = this;
    // ...

    const request = {
      url: `https://fm1.shanghainwogeng.com/api/v1/stories`,
      method: 'GET' // If no method, default is GET
    }
    // Get api data

    wx.request(request); // Then wait for response!
    // Rest of code will KEEP RUNNING while request waits!
  }
  //...
```

Not Working? -> Need WeChat permission

Wechat IDE Menu: Settings -> Project Settings:

Basic information

**Local Settings**

Project Settings

Debug Base Library

2.8.2

0.01% ▾

?

☒

Transpile ES6 to ES5

☐

Transpile Enhance

☐

Use NPM module

☐

Auto Run Audits

☒

Does not verify valid domain names, web-view (business domain names), TLS versions and HTTPS certificates

☒

The style is autocompleted when uploading code

☒

Compressed & Uglify automatically when uploading code (UglifyJs)

☐

Code protect when uploading

☐

Enable Custom Processing Commands

## **5. Receive data from response**

Add a new function in your index page called `getRequestData`

```
// /pages/index/index.js

Page({
  //...
  getRequestData: function (res) {
    console.log(res)
  },

  onLoad: function (options) {
    //...
  }
})
```



Call the function when request responds `success`

```
// /pages/index/index.js

Page({
  //...
  onLoad: function (options) {
    // Save reference to page
    let page = this;
    //...

    const request = {
      url: `https://fml.shanghaiwogeng.com/api/v1/stories`,
      method: 'GET', // If no method, default is GET
      success: page.getRequestData
    }
    // Get api data

    wx.request(request);
  }
  //...
})
```

## **6. Handle the data**

Pass the data in the response to the handler

```
// /pages/index/index.js

Page({
  //...
  getrequestData: function (res) {
    console.log(res)

    const data = res.data;
    page.setStories(data);
  },

  onLoad: function (options) {
    //...
  }
})
```

Add a new function in your index page called `setStories` to handle data

```
Page({
  //...
  setStories: function (data) {
    // Save reference to page
    let page = this;

    // Take the stories from data passed in
    const stories = data.stories;

    // Update local stories data
    page.setData({
      stories: stories
    });
  }
})
```

**2TH ENDPOINT: CREATE**

## **1. Use API token (or key)**

Not needed for open API (e.g. anyone can create, no login)

## 2. Specify endpoint

Restful: Verb and Path

```
POST /api/v1/stories
```

Combine with host => Same endpoint, but POST verb:

```
https://fm1.shanghainwogeng.com/api/v1/stories
```

```
// /pages/index/index.js
```

```
Page({
```

```
  //...
```

```
  onLoad: function (options) {
```

```
    // Save reference to page
```

```
    let page = this;
```

```
    //...
```

```
    const request = {
```

```
      url: `https://fm1.shanghainwogeng.com/api/v1/stories`,
```

```
      method: 'POST'
```

```
    }
```

```
  }
```

```
  //...
```



### **3. Attach request data**

As in a GET shown above, data is sent as a json object

For create, data comes from form submission on `post` page:

```
// pages/post/post.js

Page({
  //...

  // New Story Submission
  bindSubmit: function (event) {
    console.log(event.detail.value.name)
    console.log(event.detail.value.content)

    let name = event.detail.value.name
    let text = event.detail.value.text
  }
  //...
```

Then make form story data into request data

```
// /pages/post/post.js
// in bindSubmit

//...
let story = {
  name: name,
  text: text
}

const request = {
  url: `https://fml.shanghaiwogeng.com/api/v1/stories`,
  method: 'POST',
  data: story
}
```

4. Send request and **wait** for response

## MINI PROGRAM: New Story

```
// in Page() pages/post/post.js, New Story Submission
bindSubmit: function (event) {
  console.log(event.detail.value.name)
  console.log(event.detail.value.content)

  let name = event.detail.value.name
  let text = event.detail.value.text

  let story = {
    name: name,
    text: text
  }

  const request = {
    url: `https://fml.shanghaiwogeng.com/api/v1/stories`,
    method: 'POST',
    data: story
  }
  // Post data to API

  wx.request(request); // Then wait for response!
}
```

As before, allow WeChat permission for this api or skip the permission check.

## **5. Receive data from response**

No response data is needed, instead we'll redirect back to index

## 6. Handle the data

Redirect is called in a function called `success` in the request.

We don't need a separate page function as in `index.js`

TIP: JSON allows you to define functions inside to save you time

```
// /pages/post/post.js
// in bindSubmit

//...

const request = {
  url: `https://fml.shanghaiwogeng.com/api/v1/stories`,
  method: 'POST',
  data: story,
  success() {
    // redirect to index page when done
    wx.redirectTo({
      url: '/pages/index/index'
    });
  }
}

//...
```



## Mini Program functions with API

Read more about how Mini Programs can work with APIs

- [Wechat Doc on Network Requests \(English\)](#)

# **API resources**

What APIs can I use? Where to find them?

## Global

[rapidapi.com](https://rapidapi.com)

[programmableweb.com](https://programmableweb.com)

[apiforthat.com](https://apiforthat.com)

## China

[ai.baidu.com/ai-doc](https://ai.baidu.com/ai-doc)

[shenjian.io](https://shenjian.io)

[juhe.cn](https://juhe.cn)

## B2D - API Economy

Revenue through api's

50% => [Salesforce.com](https://www.salesforce.com)

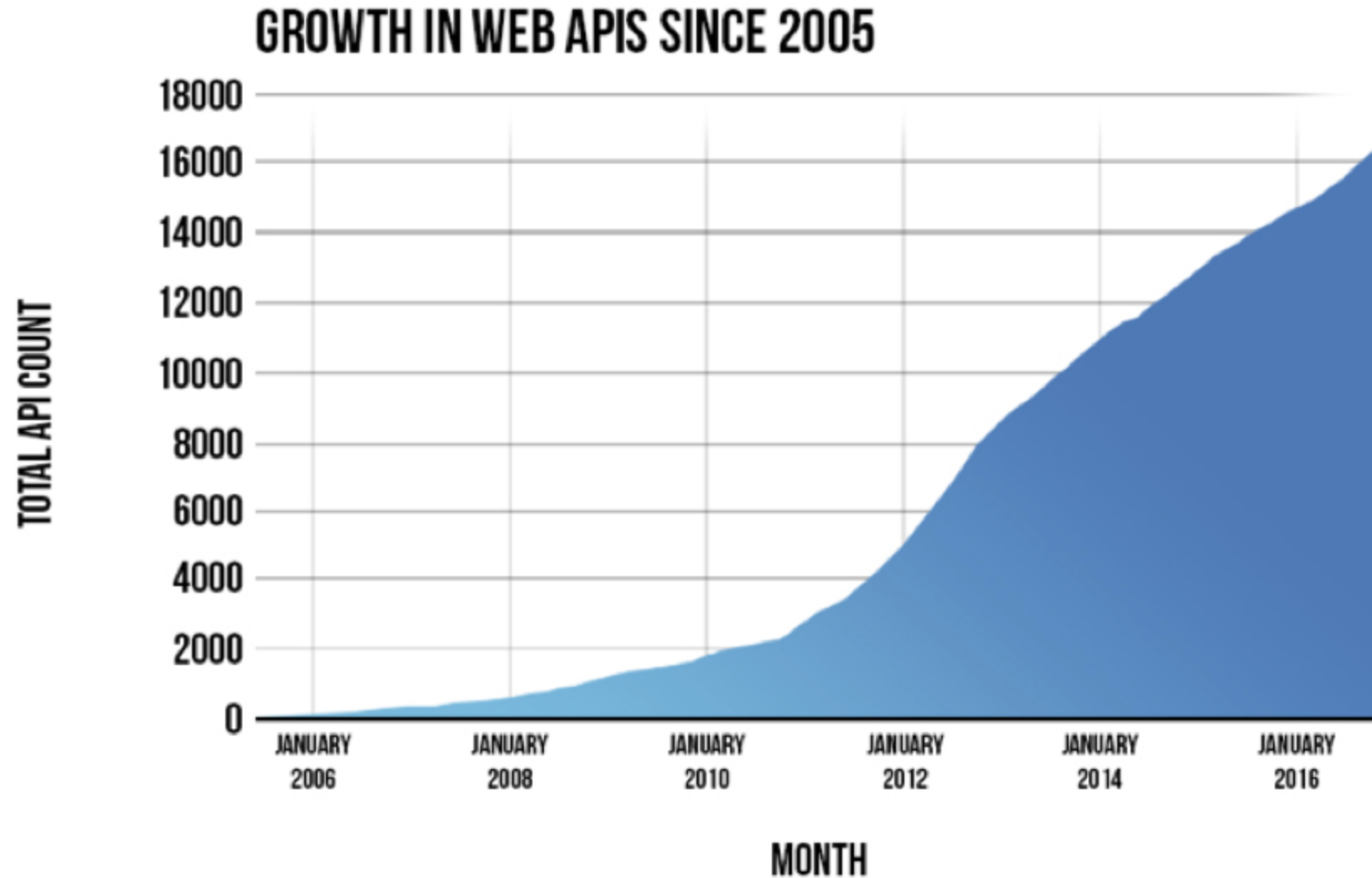
60% => [eBay.com](https://www.ebay.com)

90% => [Expedia.com](https://www.expedia.com)

# There are 17,000 APIs

9 million private api developers

Explosion of APIs growth



# API Strategy

- APIs are key to prototyping - great for entrepreneurs
- Focus on unique functionalities for business  
delivering initial product quickly & less expensively
- APIs are vital to digital transformation
- Use infrastructure so you don't build from scratch or reinvent the wheel

**HAPPY API-ING!**