API Naming Convention

1. Use hyphens (-) to improve the readability of URIs

Bad examples:

```
http://api.example.com/v1/store/vendormanagement/{vendor-id}  
http://api.example.com/v1/store/itemmanagement/{item-id}/producttype  
http://api.example.com/v1/store/inventory_management
```

Good examples:

```
http://api.example.com/v1/store/vendor-management/{vendor-id} // http://api.example.com/v1/store/item-management/{item-id}/product-type // http://api.example.com/v1/store/inventory-management // htt
```

2. Use Lowercase letters

Example: /users/{id}/pending-orders instead of /users/{id}/Pending_Orders

3. Use Nouns to represent resources / Not Verbs

Bad examples:

```
http://api.example.com/v1/store/CreateItems/{item-id}
http://api.example.com/v1/store/getEmployees/{emp-id}
http://api.example.com/v1/store/update-prices/{price-id}
http://api.example.com/v1/store/deleteOrders/{order-id}
```

Good examples:

```
http://api.example.com/v1/store/items/{item-id} 
http://api.example.com/v1/store/employees/{emp-id} 
http://api.example.com/v1/store/prices/{price-id} 
http://api.example.com/v1/store/orders/{order-id}
```

4. Use Pluralized Nouns for resources

Bad examples (Typical and Singleton resources):

http://api.example.com/v1/store/item/{item-id}
http://api.example.com/v1/store/employee/{emp-id}/address

Good examples (Typical and Singleton resources):

http://api.example.com/v1/store/items/{item-id} // http://api.example.com/v1/store/employees/{emp-id}/address

5. Use forward slashes (/) for hierarchy but not trailing forward slash (/)



6. Avoid using file extensions

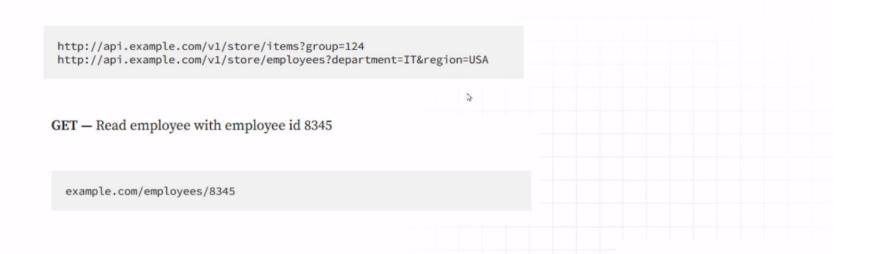
Bad examples:

http://api.example.com/v1/store/items.json
http://api.example.com/v1/store/products.xml

Good examples:

http://api.example.com/v1/store/items✓
http://api.example.com/v1/store/products✓

7. Use query component to filter URI collection



8. Avoid abridging endpoint/resource names

Example: /users/{id}/phone-number instead of /users/{id}/tel-no

Naming convention for APIs all in one!

Verb	Path	action	used for
GET	/photos	index	display a list of all photos
GET	/photos/new	new	return an HTML form for creating a new photo
POST	/photos	create	create a new photo
GET	/photos/:id	show	display a specific photo
GET	/photos/:id/edit	edit	return an HTML form for editing a photo
PUT	/photos/:id	update	update a specific photo
DELETE	/photos/:id	destroy	delete a specific photo

Some more tips!

- Consistency is key!
- Avoid Special Characters.