



`/api?type=user&id=23`



`/users/23`

Use Identifiers in URLs Instead of Query Strings

BAD!



GET /ordersByUserId/{userId}
GET /getOrder/{orderId}/byUser/{userId}

Good



GET /users/{userId}/orders
GET /users/{userId}/orders/{orderId}

Keep URLs Hierarchical and Logical

GET /getUsers
POST /createUser
PUT /updateUser/{id}
DELETE /deleteUser/{id}



GET /users
GET /users/{id}
POST /users
PUT /users/{id}
DELETE /users/{id}

Use Nouns, Not Verbs for resource name



GET /product
GET /product/{id}



GET /products
GET /products/{id}

Use Plurals for Collections



/database12345/cust_table/user_row=3345



/customers/3345

Design for Clients, Not Your Data



/customer_list/1



/customers/1

Avoid Collection Verbiage in URLs



GET /products



GET /v1/products

GET /v2/products

Use a versioning strategy in the URL to prevent breaking changes.



/customers/1/orders/8769/products/lineitems/23/details



/customers/1/orders/8769/lineitems/23

Keep URLs Short and Logical



GET /userAccounts
GET /UserAccounts



GET /user-accounts

Use Consistent Case and Formatting



/Customers/1/Orders/8769



/customers/1/orders/8769

Use Lowercase in URLs



GET

/companies/{companyId}/departments/{departmentId}/projects/
{projectId}/tasks/{taskId}



GET /projects/{projectId}/tasks
GET /projects/{projectId}/tasks/{taskId}

Avoid excessive levels of nesting; limit it to meaningful relationships.



GET /products/category/electronics/price/100-500
GET /products/page/2/limit/10/sort/price/asc



GET /products?category=electronics&price_min=100&price_max=500
GET /products?sort=price&order=asc&page=2&limit=10

Use Query Parameters for Filtering, Sorting, and Pagination