GraphQL cheatsheet

A query language for your APIs

Why GraphQL?

- No overfetching
- Type safety
- Introspection
- Self documenting
- Code generation
- Versionless design
- and more!

Learn more at **graphql.org**

Describe your data

```
type Project {
  name: String
  tagline: String
  contributors: [User]
}
```

Ask for what you want

```
{
  project(name:
"GraphQL") {
    tagline
  }
}
```

Get predictable results

```
{
  "project": {
    "tagline": "A query
language for APIs"
  }
}
```

Operations

```
query: read data
mutation: modify data
subscription: listen
to data updates
```

Built in scalars

```
scalar Int
scalar Float
scalar String
scalar Boolean
scalar ID
```

Objects

```
type User {
 id: ID
 name: String
 age: Int
}
```

Interfaces

```
interface Node {
 id: ID
}

type User implements
Node {
 id: ID
 name: String
 age: Int
}
```

Unions

```
union Sponsor = User |
Organisation
```

Enums

```
enum Visibility {
 PUBLIC
 INTERNAL
 PRIVATE
}
```

Custom scalars

```
scalar DateTime
scalar Url
```

Arguments

```
type Query {
 user(id: ID): User
}
```

Input objects

```
input UserInput {
 name: String
 age: Int
}

type Mutation {
 updateUser(userInput:
UserInput): UserResult
}
```

Nullability

```
type User {
 # non-nullable
 name: String!
 # nullable
 email: String
}
```

Arrays

```
type User {
 projects: [Project!]!
 friends: [User]
}
```

Descriptions

```
“A user of this API”
type User {
 “The name of the
user”
 name: String
}
```

Comments

```
# comments start with
# the ‘#’ character
```