API Design

Divyansh Rajesh Jain

UCDAVIS

What is an API?

- API → Application Programming Interface
- Allows for there to be modularity
- “Contract” between two softwares
- Examples
    - REST
    - GraphQL

Why we care about API Design?

- Affects maintainability of software
- Allows for clean distribution of workload across software (Modularity)
- Encourages “third party” integration which can improve User Experience

How to Design Good APIs?

- Simplicity
- Consistency
- Flexibility
- Performance

Simplicity in API Design

- Only include functionality which is essential to the application
- Try to abstract away as much complexity from the user
- Provide common interfaces to similar features even if the backend is different

Consistency in API Design

- Naming conventions are the same across all functions
- Structure of data which is returned to the user is the same
- Semantics of API are consistent across operations

Flexibility

- Provide a level of abstraction which can handle many use cases with a common interface
- Important design principle when building private API to “wrap” other services

Performance

- API should be extensible
    - Can “scale” → Performance-wise or support wise
- Part of performance is also maintainability
    - Should be “easy” to maintain

Different Type of APIs

- System Level APIs (System Calls)
- Web Interface APIs
  - REST
  - GraphQL
- Library written to wrap other services

Thank You!

Next Time: More on API Design (Hands on Activity)