clang-extract-api
Clang support for API information generation in JSON

Zixu Wang

2022 LLVM Developers’ Meeting | Apple, Inc. | November 8, 2022
/// A color value with red, green, blue, and alpha components.

typedef struct Color {
    unsigned Red;
    unsigned Green;
    unsigned Blue;
    unsigned Alpha;
} Color;

/// Add opacity to a given color.
///
/// - Parameters:
///   - C: The color to modify.
///   - Opacity: The amount of opacity to add.

void addOpacity(Color *C, unsigned Opacity);
Check for API-breaking changes

Documentation generation

...
• clang -cc1 -ast-dump

• More for compiler engineers with Clang AST details
Background

Existing approaches

- Doxygen
  - Focused on documentation
  - Output (HTML/LaTeX/XML/...) not friendly for downstream tools
clang-extract-api

• A new Clang library ExtractAPI
• A new frontend action

```
clang  -extract-api  \
  -x  c-header  \
  headers/coolAPI.h  headers/anotherCoolAPI.h  \
  -isysroot  <SDK>  \
  -Iheaders  \
  -product-name=MyCoolAPIs  \
  -o  APIInfo.json
```

• Ready to be used as a library from libclang for more integrations
clang-extract-api
Design and implementation

- Working on headers
  - APIs are declared and shipped in headers
  - Sufficient and concise from a client’s perspective
  - Independent of a full build, faster and more flexible
  - `-x c-header header1.h header2.h ...`
clang-extract-api
Design and implementation

• Input headers `#include'd in a memory buffer file for parsing
• `ExtractAPIVisitor` visits the AST and collects API information
  • Macro definitions handled by `PPCallbacks`
• Finally serialization
clang-extract-api
Output format - Symbol Graph

• Represents APIs as a directed graph
• Nodes are declarations
  • Name, kind, function signature, etc.
• Edges are relationships
  • `memberOf`, `inheritsFrom`, etc.
• Language-agnostic

https://github.com/apple/swift-docc-symbolkit

```json
{
  ... (metadata)
  "symbols" : [
    {
      "name" : "Color",
      "precise" : "c:@S@Color",
      "kind" : "Struct",
      ...
    },
    ...
  ],
  "relationships" : [
    {
      "kind" : "memberOf",
      "source" : "c:@S@Color@FI@Red",
      "target" : "c:@S@Color"
    },
    ...
  ]
}
```
Usage Showcase
Xcode/Swift-DocC Documentation for C/Objective-C

• Swift-DocC is an open-source documentation compiler
• Build rich documentation for Swift and Objective-C projects
• Integrated in Xcode to display right in the Developer Documentation Window
• Standalone tool to host documentation on a website

https://www.swift.org/documentation/docc/
https://forums.swift.org/t/clang-support-for-objective-c-symbol-graph-generation/
/// A color value with red, green, blue, 
/// and alpha components.

typedef struct Color {
    unsigned Red;
    unsigned Green;
    unsigned Blue;
    unsigned Alpha;
} Color;

/// Add opacity to a given color.
///
/// - Parameters:
///   - C: The color to modify.
///   - Opacity: The amount of opacity to add.
void addOpacity(Color *C, unsigned Opacity);

$ clang -extract-api \
-x c-header include/Color.h \
-Iinclude \
-product-name=Color \
-o Color.symbols.json

$ docc preview \
  --fallback-display-name Color \
  --fallback-bundle-identifier demo.Color \
  --additional-symbol-graph-dir .
Usage Showcase
Xcode/Swift-DocC Documentation for C/Objective-C

Framework
Color

Topics
Structures
Color
A color value with red, green, blue, and alpha components.

Functions
addOpacity
Add opacity to a given color.

Structure
Color
A color value with red, green, blue, and alpha components.

Declaration

void addOpacity(Color *C, unsigned int Opacity)

Parameters
C
The color to modify.

Opacity
The amount of opacity to add.
Future Directions

• More use cases
• Support for C++
• More custom serializers
• ...

Followup References

• LLVM Discourse update on clang-extract-api:
  • https://discourse.llvm.org/t/update-on-clang-extract-api-clang-support-for-api-information-generation-in-json/

• Symbol Graph: https://github.com/apple/swift-docc-symbolkit/

• Swift-DocC: https://www.swift.org/documentation/docc/

• clang-extract-api support for Swift-DocC:
  • https://forums.swift.org/t/clang-support-for-objective-c-symbol-graph-generation/

• Example and instructions to try out
Thanks!