clang-extract-api

Clang support for API information generation in JSON

Zixu Wang

2022 LLVM Developers' Meeting | Apple, Inc. | November 8, 2022

Background

API information

- Name, type, function signature
- Documentation comments
- Attributes (availability, …)
- Relationships with other APIs

```
/// 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);
```

Background

Uses of API information

- Check for API-breaking changes
- Documentation 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);
```

Background

Existing approaches

- `clang -cc1 -ast-dump`
    - More for compiler engineers with Clang AST details

```
...
|-RecordDecl 0x7fe4f382b400 <include/Color.h:3:9, line:8:1> line:3:16 struct Color definition
| |-FullComment 0x7fe4f382bb10 <line:1:4, line:2:25>
| | `-ParagraphComment 0x7fe4f382bae0 <line:1:4, line:2:25>
| |   |-TextComment 0x7fe4f382ba90 <line:1:4, col:40> Text=" A color value with red, green, blue,"
| |   `-TextComment 0x7fe4f382bab0 <line:2:4, col:25> Text=" and alpha components."
| |-FieldDecl 0x7fe4f382b4b8 <line:4:3, col:12> col:12 Red 'unsigned int'
| |-FieldDecl 0x7fe4f382b520 <line:5:3, col:12> col:12 Green 'unsigned int'
| |-FieldDecl 0x7fe4f382b588 <line:6:3, col:12> col:12 Blue 'unsigned int'
| `-FieldDecl 0x7fe4f382b5f0 <line:7:3, col:12> col:12 Alpha 'unsigned int'
|-TypedefDecl 0x7fe4f382b698 <line:3:1, line:8:3> col:3 referenced Color 'struct Color':'struct Color'
| |-ElaboratedType 0x7fe4f382b640 'struct Color' sugar
| | `-RecordType 0x7fe4f382b480 'struct Color'
| |   `-Record 0x7fe4f382b400 'Color'
| `-FullComment 0x7fe4f382bc00 <line:1:4, line:2:25>
|   `-ParagraphComment 0x7fe4f382bbd0 <line:1:4, line:2:25>
|     |-TextComment 0x7fe4f382bb80 <line:1:4, col:40> Text=" A color value with red, green, blue,"
|     `-TextComment 0x7fe4f382bba0 <line:2:4, col:25> Text=" and alpha components."
`-FunctionDecl 0x7fe4f382b958 <line:15:1, col:43> col:6 addOpacity 'void (Color *, unsigned int)'
  |-ParmVarDecl 0x7fe4f382b7c0 <col:17, col:24> col:24 C 'Color *'
  |-ParmVarDecl 0x7fe4f382b840 <col:27, col:36> col:36 Opacity 'unsigned int'
  `-FullComment 0x7fe4f382bd70 <line:10:4, line:14:46>
    |-ParagraphComment 0x7fe4f382bca0 <line:10:4, col:33>
    | `-TextComment 0x7fe4f382bc70 <col:4, col:33> Text=" Add opacity to a given color."
    `-ParagraphComment 0x7fe4f382bd40 <line:12:4, line:14:46>
      |-TextComment 0x7fe4f382bcc0 <line:12:4, col:17> Text=" - Parameters:"
      |-TextComment 0x7fe4f382bce0 <line:13:4, col:31> Text="   - C: The color to modify."
      `-TextComment 0x7fe4f382bd00 <line:14:4, col:46> Text="   - Opacity: The amount of opacity to add."
```

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

```
{
  ... (metadata)
  "symbols" : [
    {
      "name" : "Color",
      "precise" : "c:@S@Color",
      "kind" : "Struct",
      ...
    },
    ...
  ],
  "relationships" : [
    {
      "kind" : "memberOf",
      "source" : "c:@S@Color@FI@Red",
      "target" : "c:@S@Color"
    },
    ...
  ]
}
```

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

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/

Usage Showcase

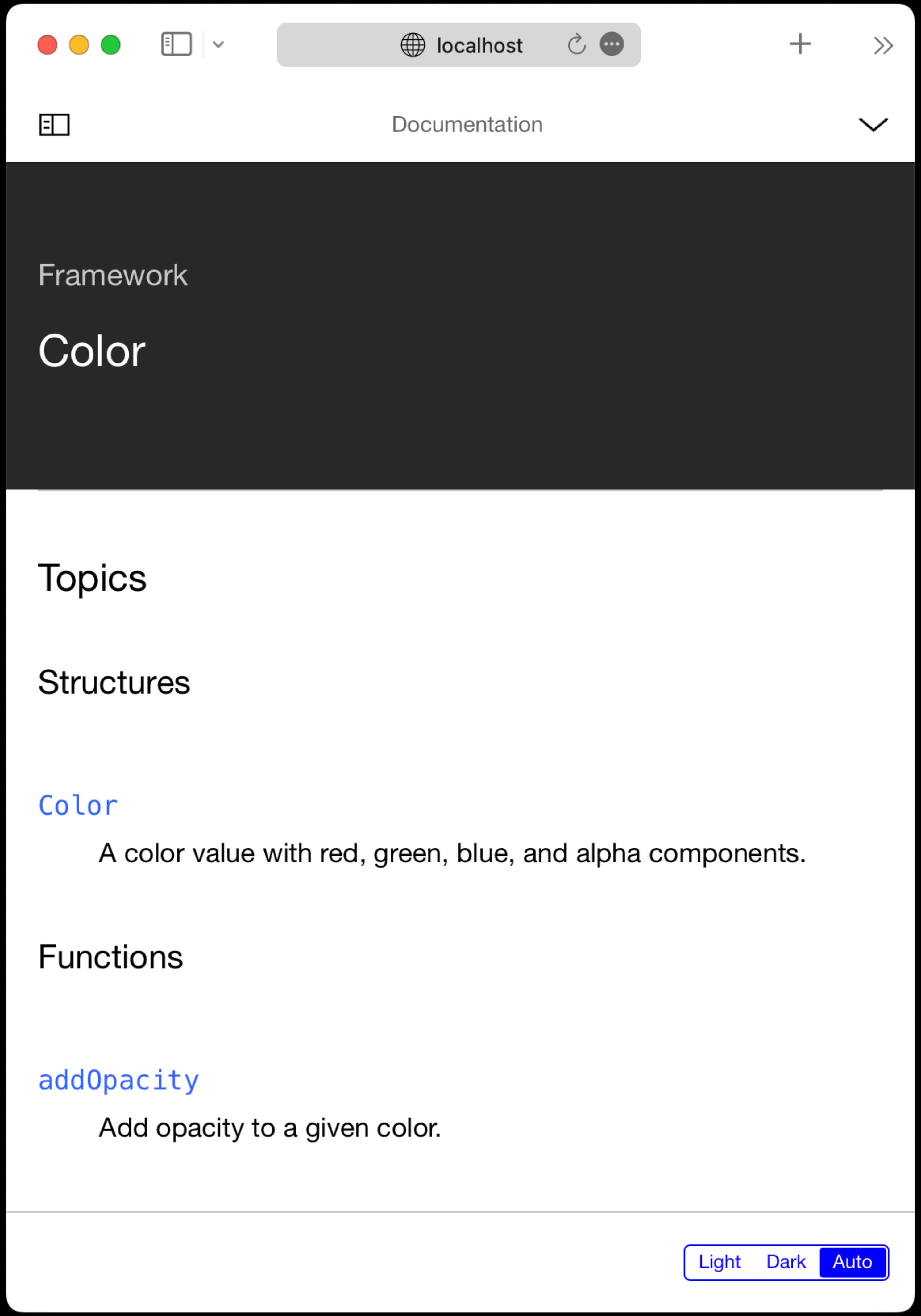
Xcode/Swift-DocC Documentation for C/Objective-C

```c
/// 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


localhost

Documentation

Framework

Color

Topics

Structures

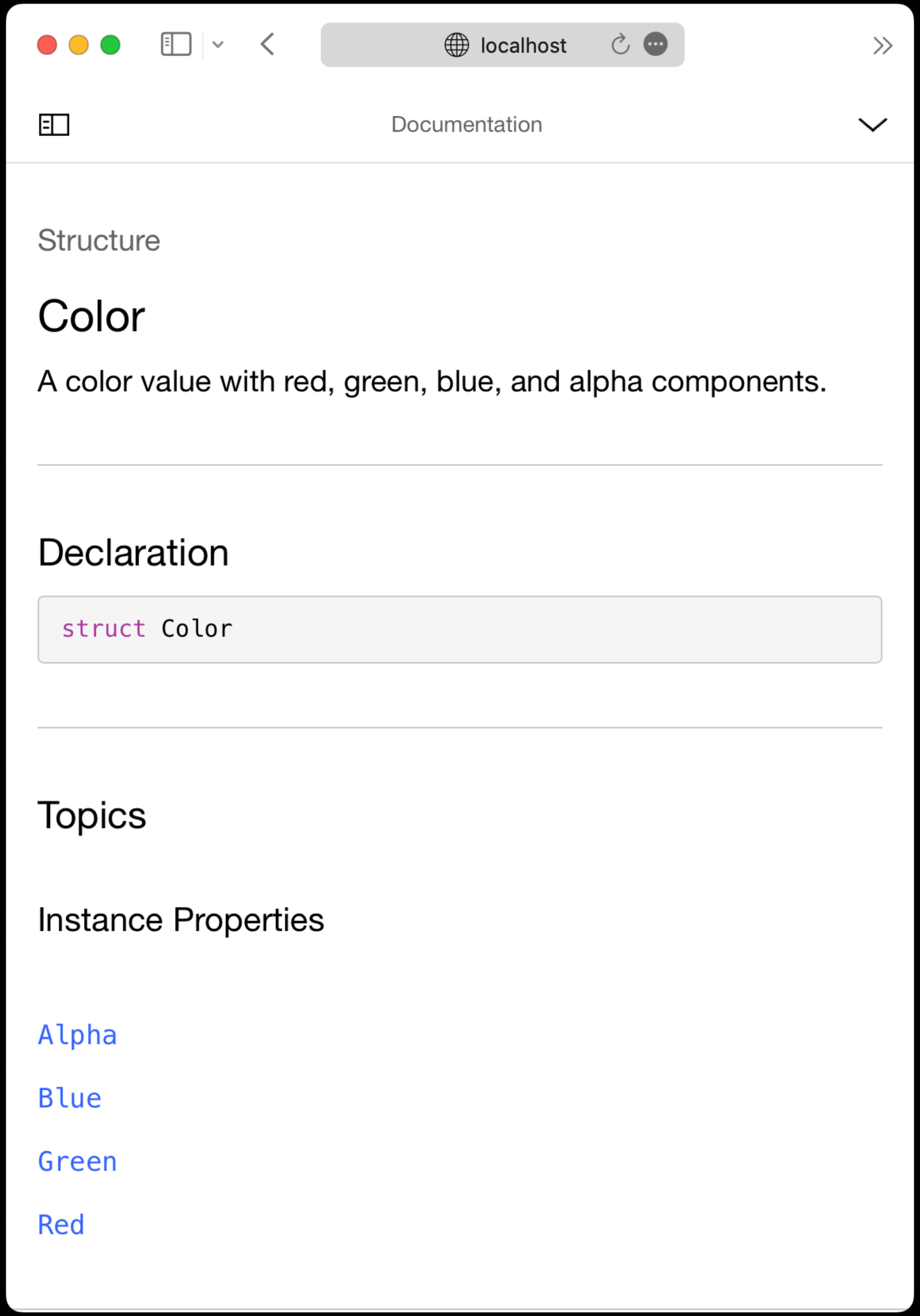
`Color`

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

Functions

`addOpacity`

Add opacity to a given color.


localhost

Documentation

Structure

Color

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

Declaration

```
struct Color
```

Topics

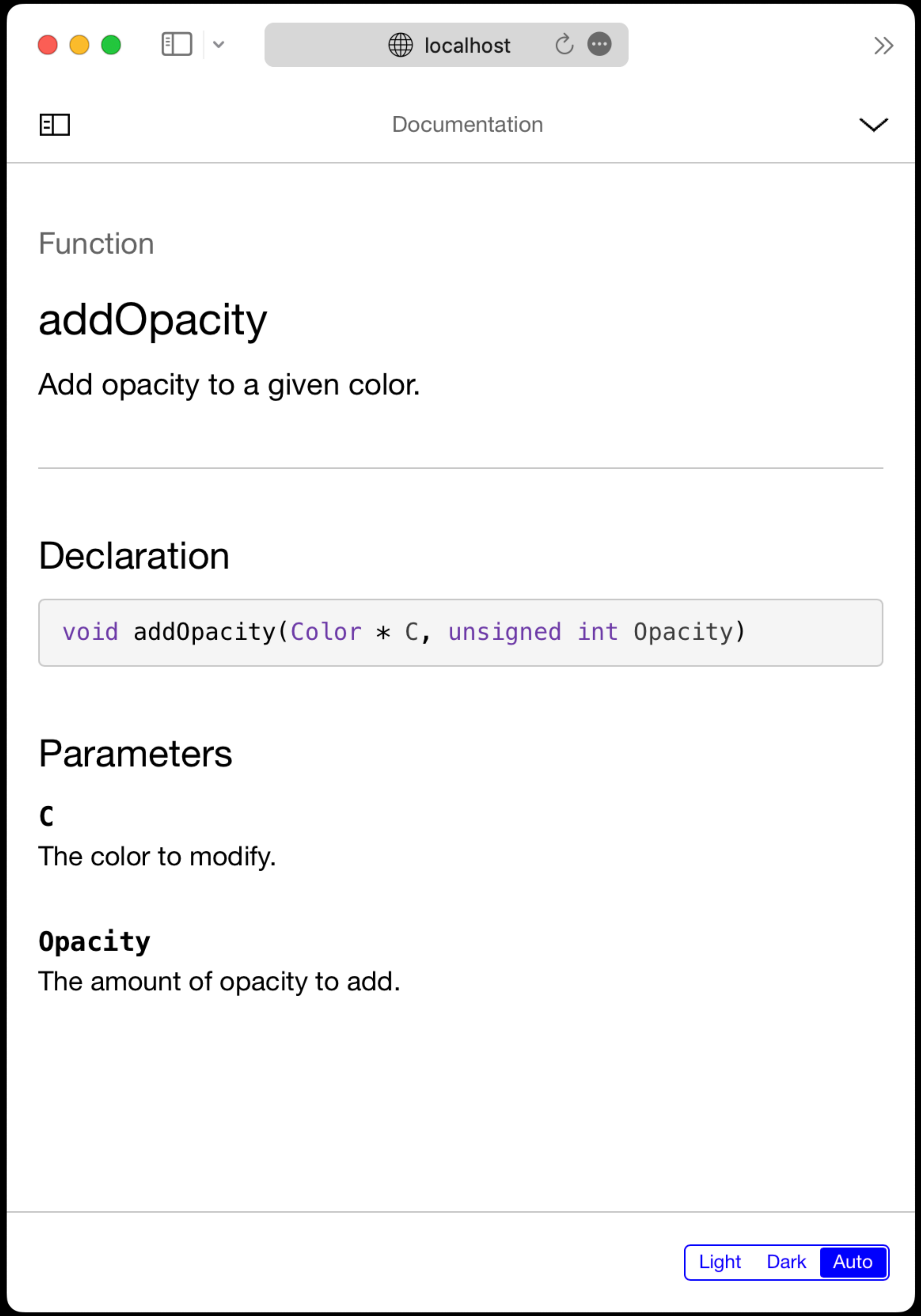
Instance Properties

`Alpha`

`Blue`

`Green`

`Red`


localhost

Documentation

Function

addOpacity

Add opacity to a given color.

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!