

[WIP] CWCTMA xMD v1.0 language documentation

What this project is for

This language is for CWCT/MA web application platform, to make dynamic webapp creation much easier. It provides handy tools for devs to make dynamic webapps using markdown based files

How does the language work?

As the name suggest, this language is based on markdown and flavored with XML and inline cshtml-like elements

Page types - coders perspective

There are n types of xmd available as of CWCTMA-XMD 1.0

Type #1 - static

This is a plain markdown file with metadata inside it

Example code:

```
<!DOCTYPE cwctma-xmd>
<metadata type="1" lang="en" isdisplayed="true">
  <id></id>
  <title></title>
  <keywords></keywords>
  <description></description>
  <search></search>
  <image alt=""></image>
  <releasedat></releasedat>
  <modifiedat></modifiedat>
</metadata>
# Demo page
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quasi eligendi laborum,
ab deleniti ipsam tempora!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quibusdam possimus,
soluta fuga dolorum architecto nesciunt, tempore mollitia delectus ipsa, eum
repudiandae animi laudantium! Non quasi itaque quod nisi, atque reprehenderit
culpa nesciunt temporibus quia consequatur, quidem iusto! Minus, error fugiat
voluptatum repellat aspernatur facere itaque placeat necessitatibus qui aperiam
tenetur.
```

Type #2 - dynamic

This is a C# flavored markdown file with metadata inside it.

Inline C# code - returns with value by default

```
@cs{#"String goes here"#}
```

```
@cs{# $"It can also contain {interpolated} string"#}
```

Normal C# code - you need to have a return value

```
@cs{>var code = "goes here and it also needs a "; var return = code; return value;
<}
```

```
@cs{>
  var it = "can be multiline, but do not forget the ";
  var return = it;
  return value; // Also you can make comments if you so desire, but why tho?!
<}
```

Example code:

```
<!DOCTYPE cwctma-xmd>
<metadata type="2" lang="en" isvisible="true">
  <id></id>
  <title></title>
  <keywords></keywords>
  <description></description>
  <search></search>
  <image alt=""></image>
  <releasedat></releasedat>
  <modifiedat></modifiedat>
</metadata>
# Demo page

@cs{# $"This is a test code. The current time is {DateTime.Now:O}"#}

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quasi eligendi laborum,
ab deleniti ipsam tempora!

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quibusdam possimus,
soluta fuga dolorum architecto nesciunt, tempore mollitia delectus ipsa, eum
repudiandae animi laudantium! Non quasi itaque quod nisi, atque reprehenderit
culpa nesciunt temporibus quia consequatur, quidem iusto! Minus, error fugiat
```

```
voluptatum repellat aspernatur facere itaque placeat necessitatibus qui aperiam  
tenetur.
```

Page types - designers perspective

All of the contents inside `><` after design elements (without any space) considered as parameter fields. Parameters are separated by `;`. In multiline contents, you only need to set up parameters at the first line.

Alert boxes

You can use them by simply putting an `]` before the content, that you make an alert of. This can also be multiline by adding `]` before each line. You can use markdown elements inside alerts.

Parameters (1 parameter available)

- info
- success
- warning
- danger

Example:

```
]>type< content goes here  
  
]>type< # Multiline  
] content  
] looks  
] like  
] this
```