

SDC19

Where Now Meets Next

Bixby Building Blocks

To Save Time And Space

Sarah Moliner-Roy

Research Engineer

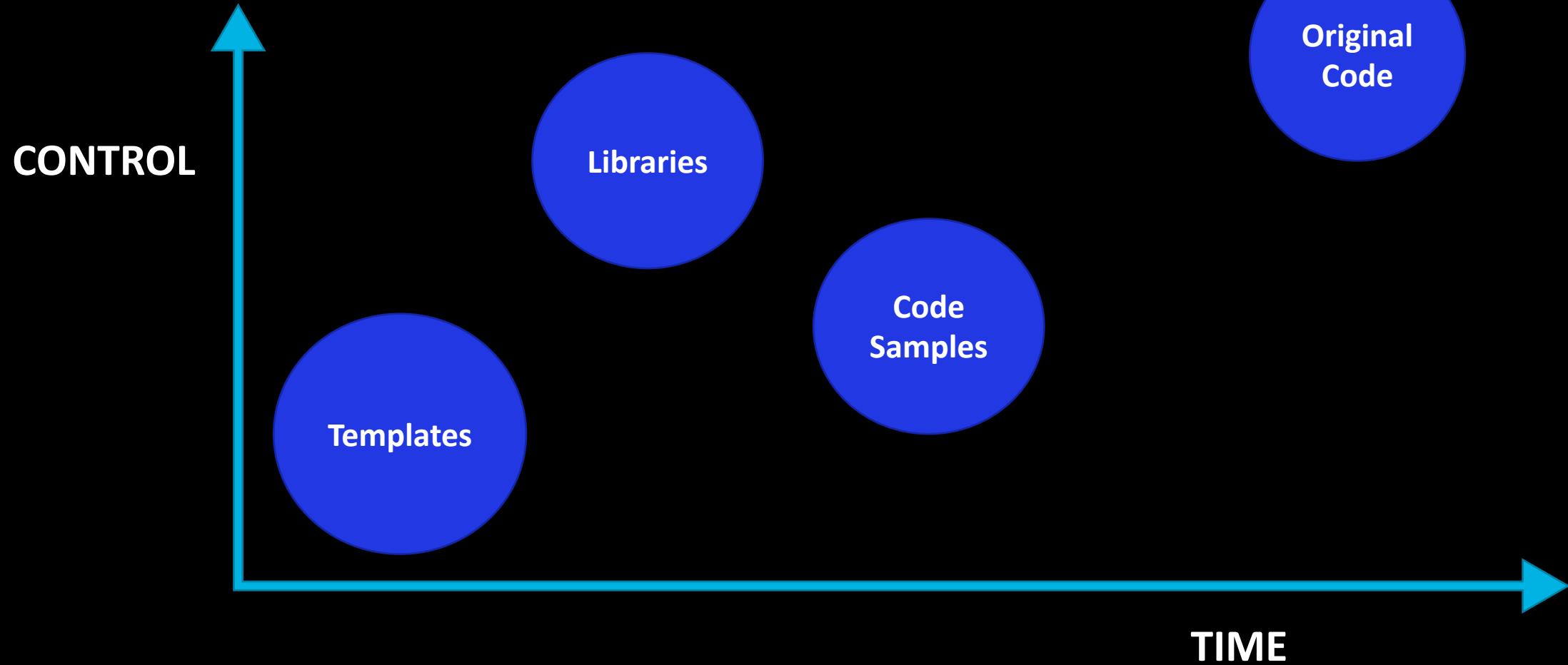
Viv Labs / Samsung Electronics



Table of Contents

- ✗ Bixby Templates
 - ✗ Facts, Jokes, Quiz, Podcast, Search
- ✗ Library Capsules
 - ✗ Time, Geo, Profile, Core and More!
- ✗ Sample Capsules
 - ✗ Audio, Views, Bank, Movie Agent...

Capsule Development





The Power of Bixby Templates

Bixby Templates

A wizard to help create new Capsules

- Choose user experience pattern
- Customize parameters
- Generate complex code
- Start from a functional end-to-end state
- Hit the ground running



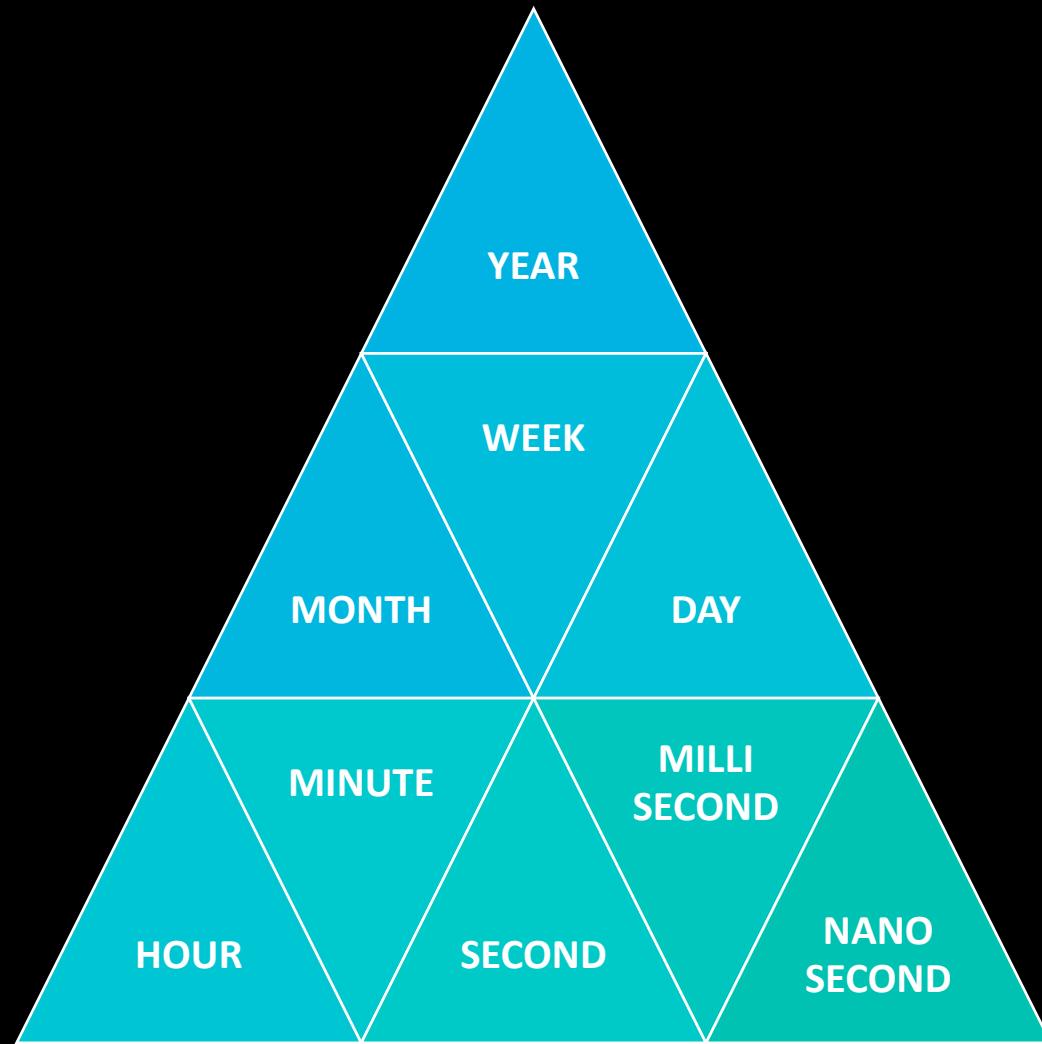
#SDC19



The Power of Library Capsules

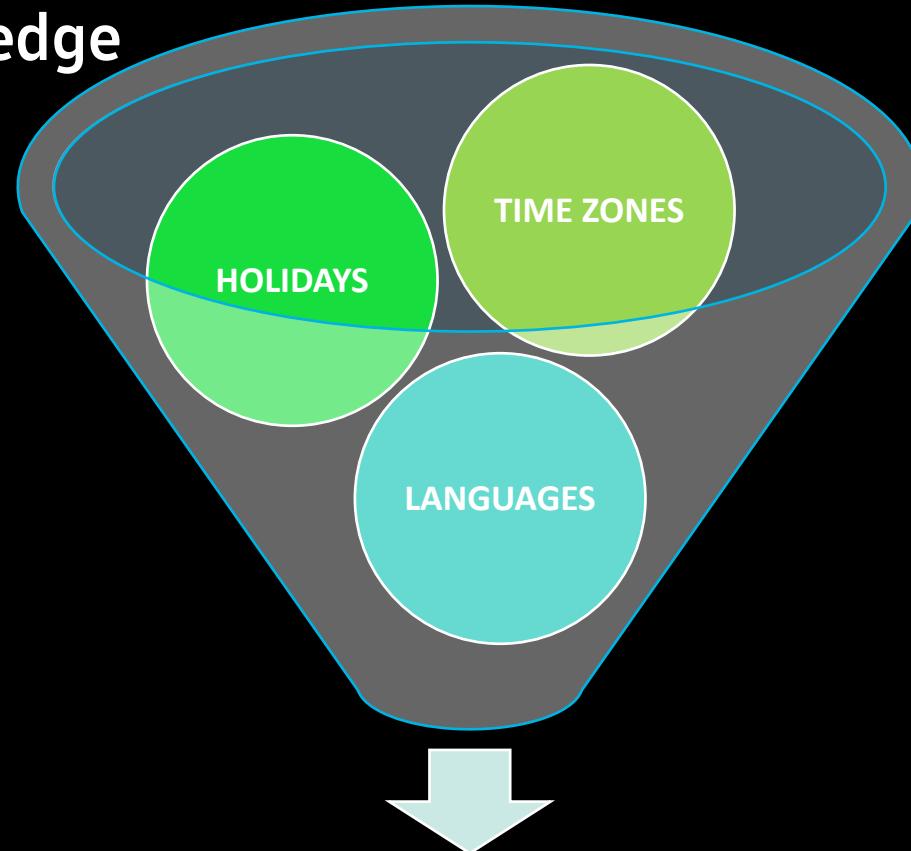
Why Use Library Capsules?

Building Blocks



Why Use Library Capsules?

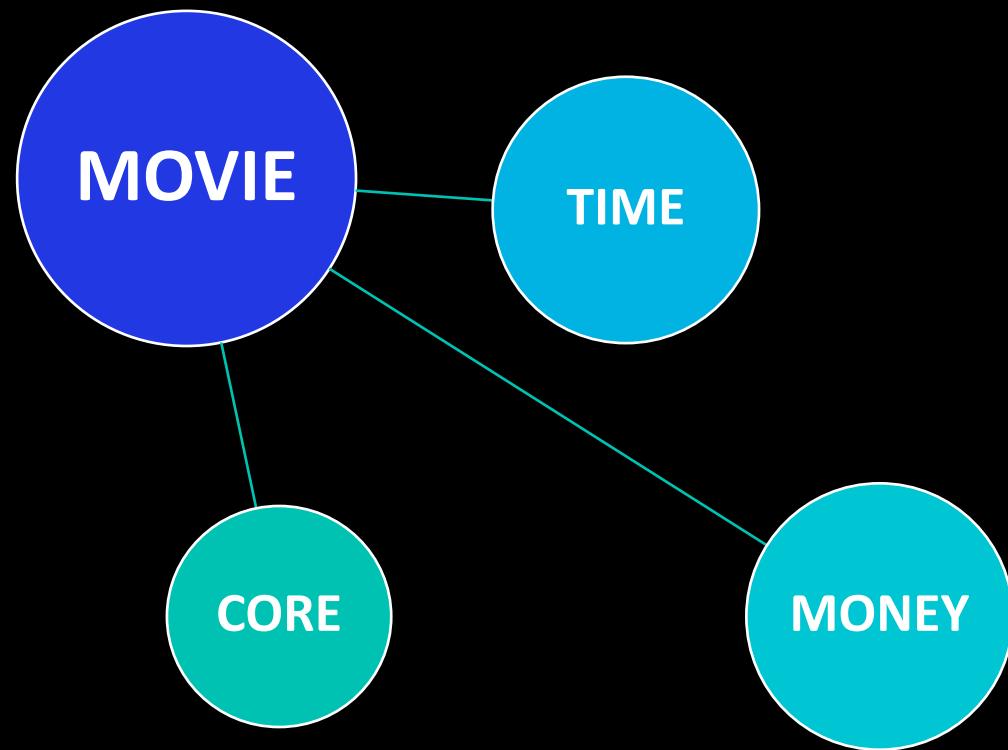
Expert Domain Knowledge



"Le dimanche de Pâques" -> "Easter Sunday" -> "The Sunday following the first full moon of spring" -> "April 12th 2020"

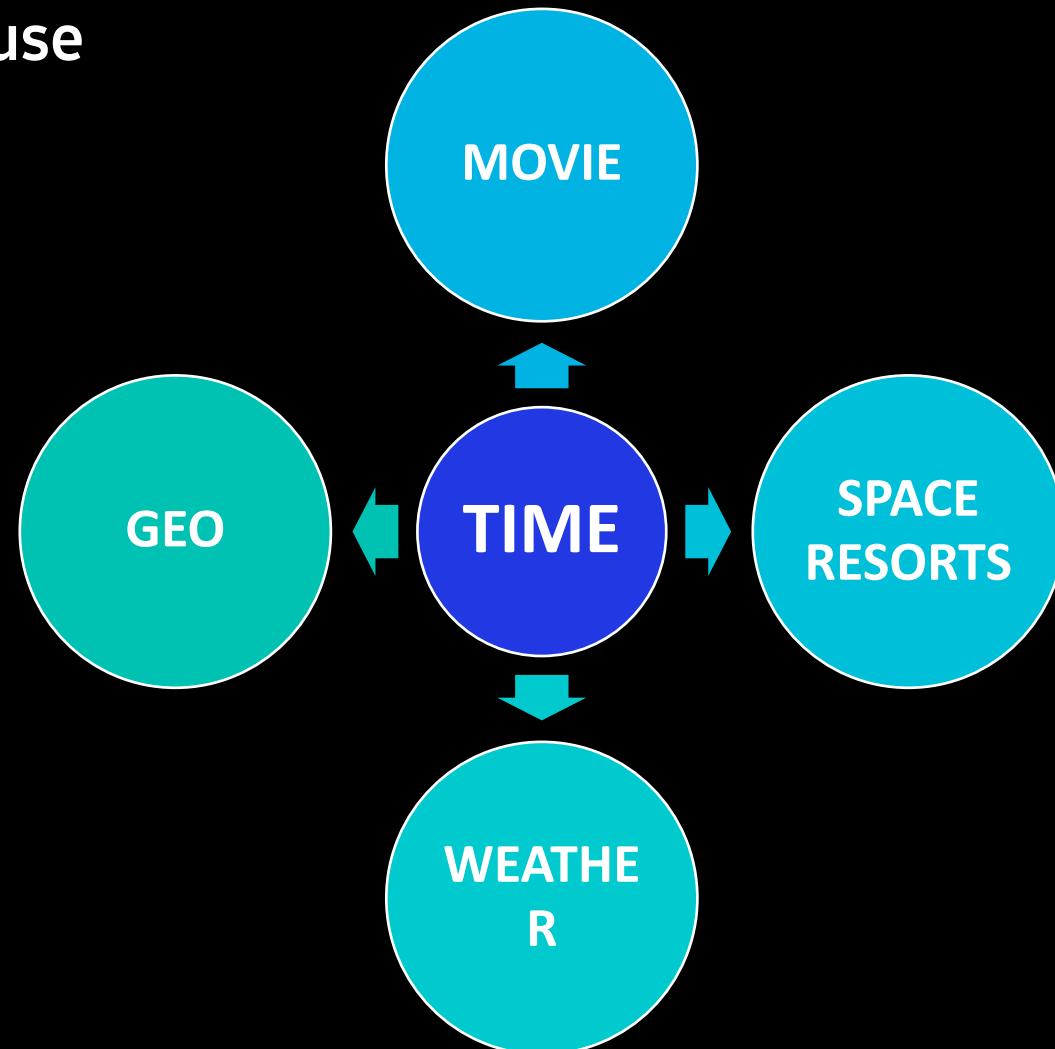
Why Use Library Capsules?

Separation Of Concerns



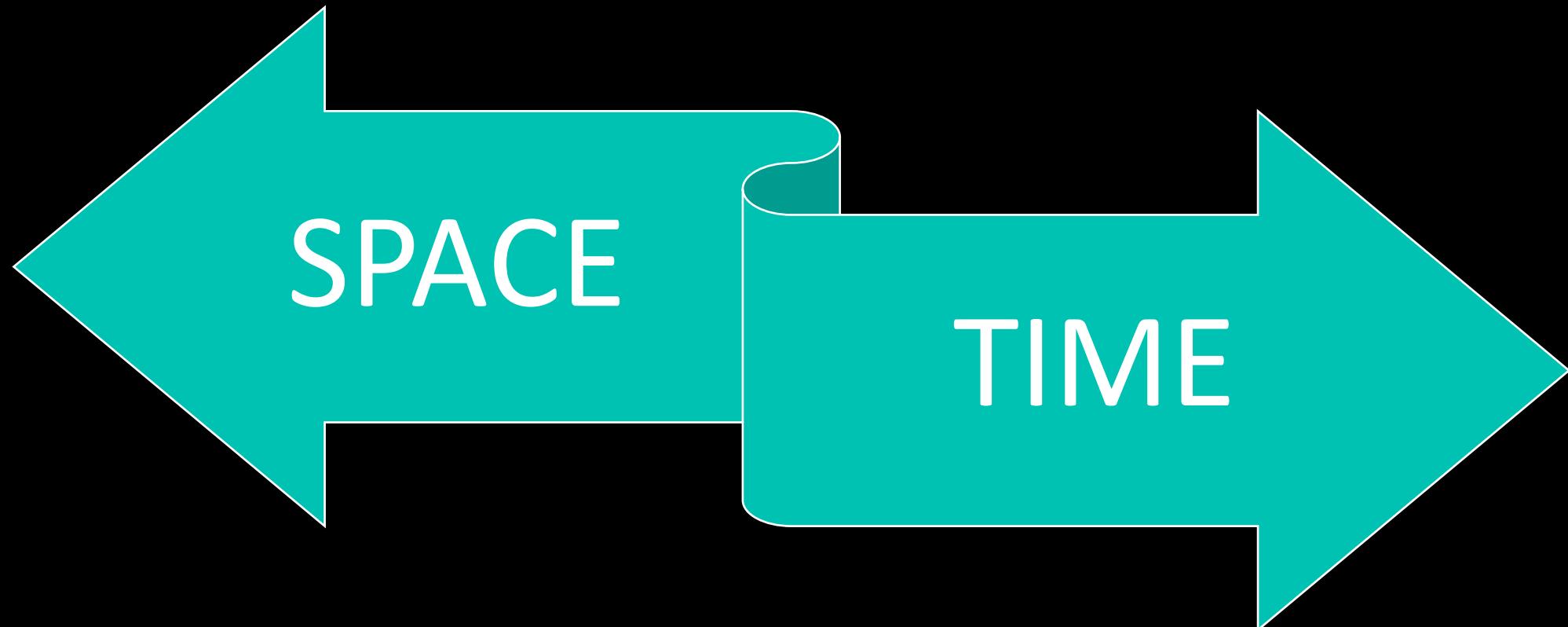
Why Use Library Capsules?

Ease Of Use And Reuse



Why Use Library Capsules?

Save Space And Time



What Are Library Capsules?

	End User Capsule	Library Capsule
Audience	End users	Capsule developers
Distribution	Published to the Bixby Marketplace	Shared and documented in the Bixby Developer Center
Access	Invoked directly by talking to Bixby	Imported in code by other Capsules
Scope	Broad and action oriented	Narrow and deep
NL	Full	Specialized only

How To Use Library Capsules?

Documentation

<https://bixbydevelopers.com/dev/docs/dev-guide/developers/library>

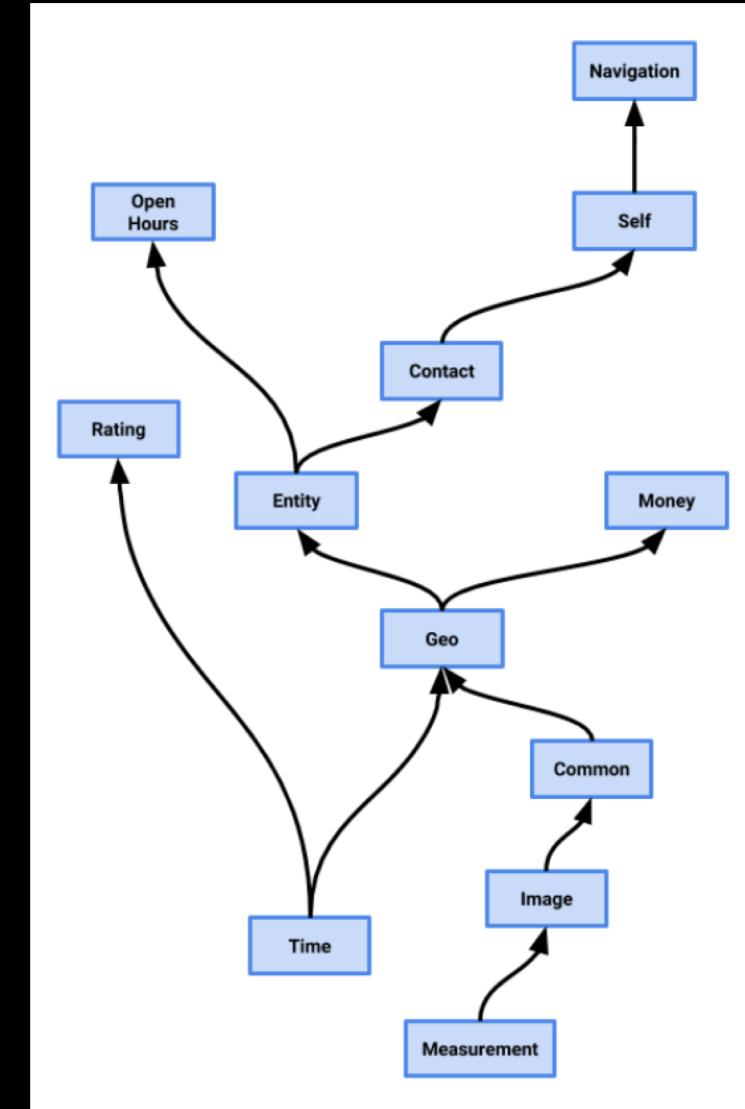
Library Capsule	Version
bixby.audioPlayer	0.3.6
viv.common	3.32.18
viv.contact	2.5152.2
viv.entity	3.20.5
viv.geo	9.20.3
viv.image	2.12.10
viv.location	1.0.18
viv.measurement	2.14.6
viv.money	2.22.3
viv.navigation	1.9.19
viv.openHours	4.1.14
viv.rating	3.2.10
viv.self	3.5.141
viv.time	2.92.45

How To Use Library Capsules?

Import and Alias

```
capsule {  
    ...  
    capsule-imports {  
        import (viv.geo) {  
            version(9.20.3)  
            as (geo)  
        }  
        import (viv.time) {  
            version-from (geo)  
            as (time)  
        }  
    }  
}
```

#SDC19



How To Use Library Capsules?

Request Permissions

```
capsule {  
    permissions {  
        user-profile-access  
    }  
}
```

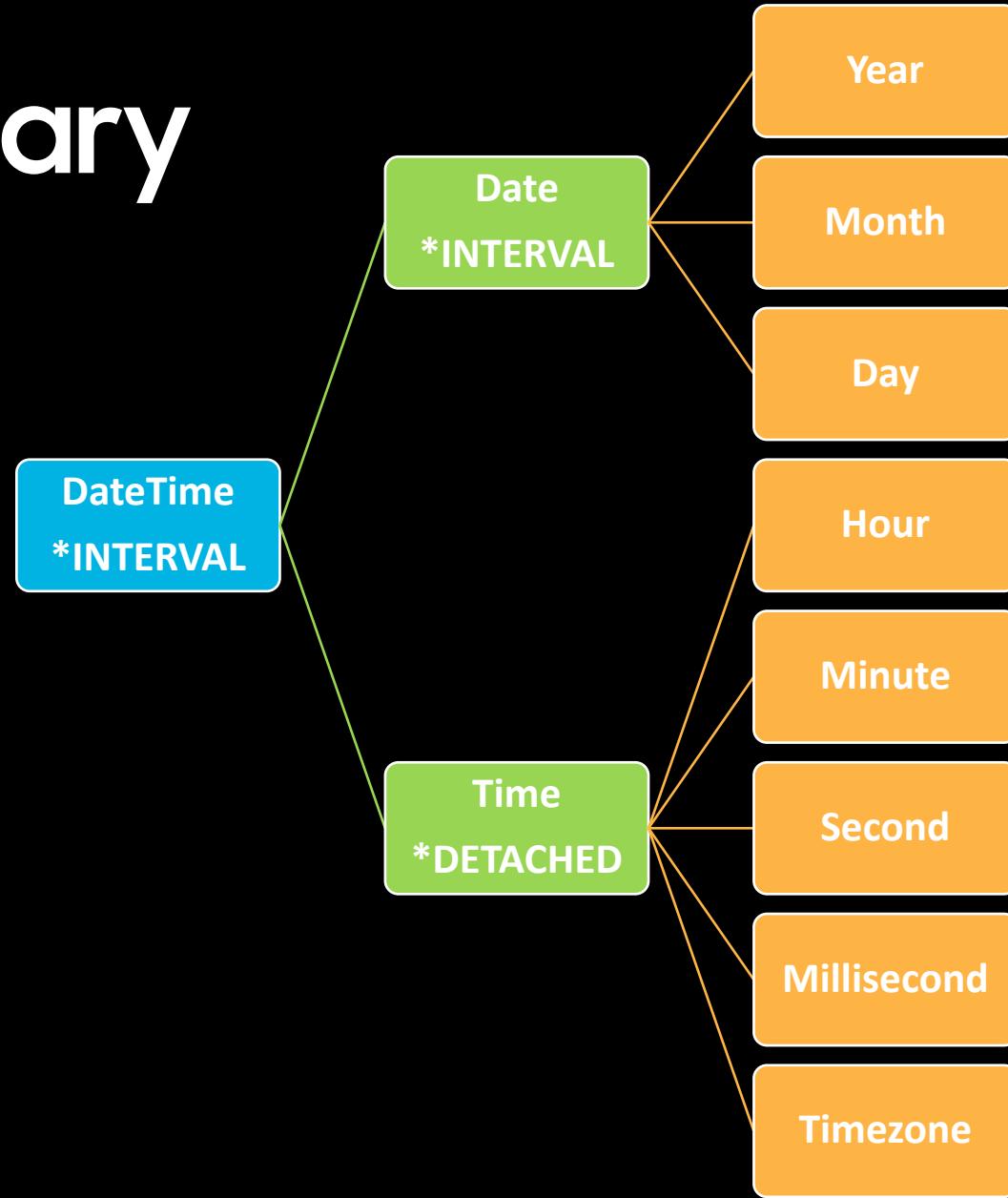
How To Use Library Capsules?

Use everywhere!

- Models
- Code
- Training
- Views
- Dialogs
- ...

Time Library

Models



Time Library

One Model To Rule Them All

Date

- Today
- Tomorrow
- Halloween
- Thursday October 31st

DateTime

- Tomorrow at 8pm
- At 7am on November 2nd
- Midnight on Friday

DateTimeExpression

DateInterval

- This week
- Next month
- In June
- 2001

DateTimeInterval

- Tonight
- This weekend
- After lunch
- Monday morning



#SDC19

Time Library

Models	Time, Date, DateTime, Intervals	Tomorrow at 7pm
Action Input	DateTimeExpression	Tomorrow at 7pm
NL Training	DateTimeExpression	Tomorrow at 7pm
Views	Calendar Time Picker	
Action Implementation	JS time module	require('dates'), serialize/deserialize, functions
Modeling	Roles, Properties	Runtime, ReleaseYear
Expression Language	Formatting functions Time functions	value(<time model>), dateTime(dateTime, pattern) now, isFuture, isPast, durationBetween

Geo Library

Models	AdministrativeDivision Locality, Neighborhood GeoPoint, Place	USA, California Los Angeles, Hollywood Current location, Hollywood Sign, LAX
Action Input	NamedPoint SearchRegion	MGM Studios 245 N Beverly Dr Within 10 miles, In Anaheim
NL Training	SearchTerm CurrentLocation	Universal Studios, Hobbiton Nearby
Views	Location auto-complete Map	Disn... Disneyland! Map with pins
Action Implementation	Javascript	latitude, longitude, radius name, locality, state, country
Modeling	Roles, Properties	Destination, Business address
Expression Language	Formatting functions Geo functions	value(<geo model>) \$user.currentLocation, calculateDistance, within

Profile Library

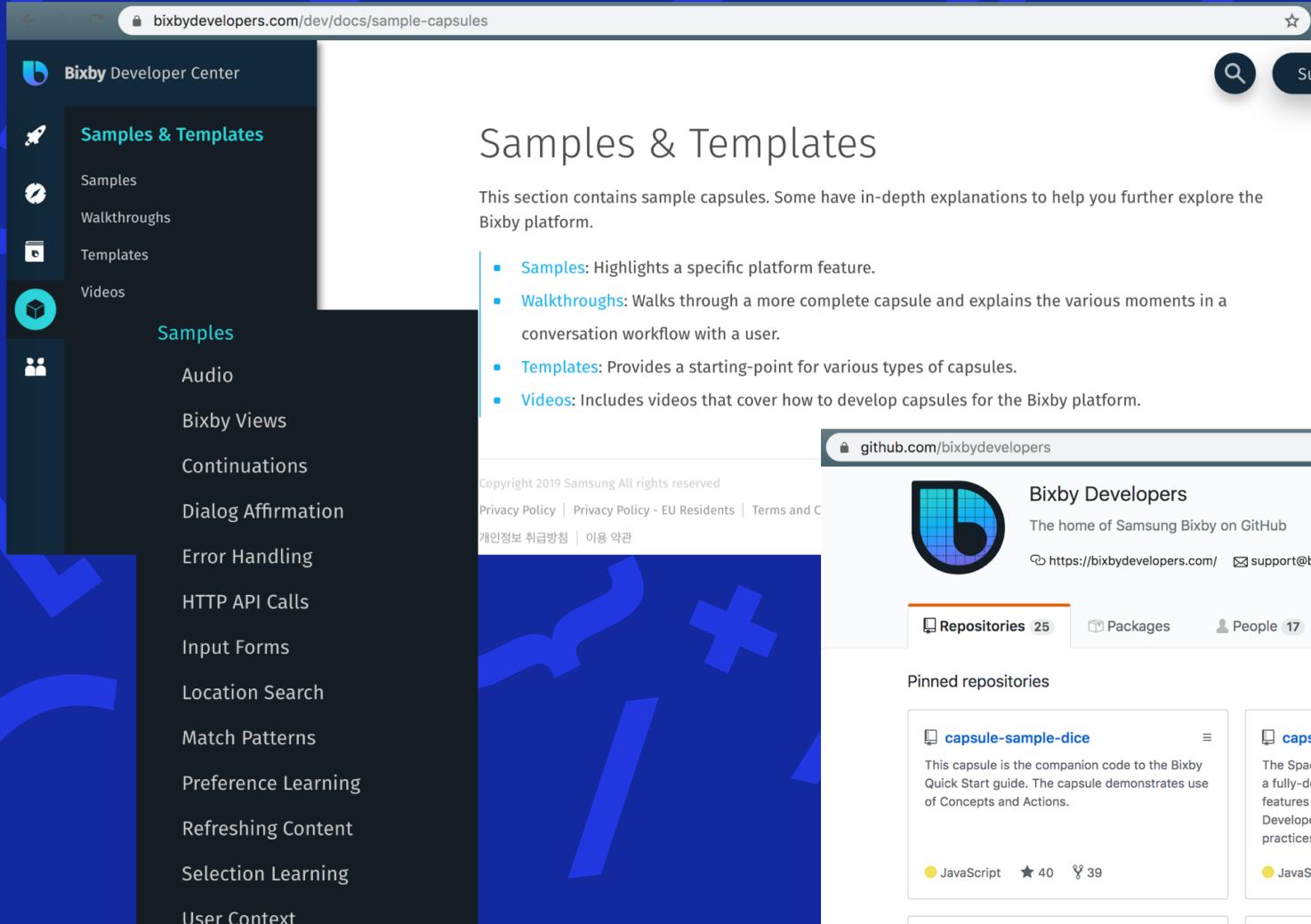
Models	Name, Email, Phone Field	Audrey Tautou firstName
Action Input	Profile	intent: goal: GetProfile
NL Training	Field AddressType, EmailType, PhoneType	change my name home, work
Views	Input form	Fill in your contact information
Action Implementation	Javascript	firstName
Modeling	Role	role-of(Profile)
Expression Language		

Other Libraries

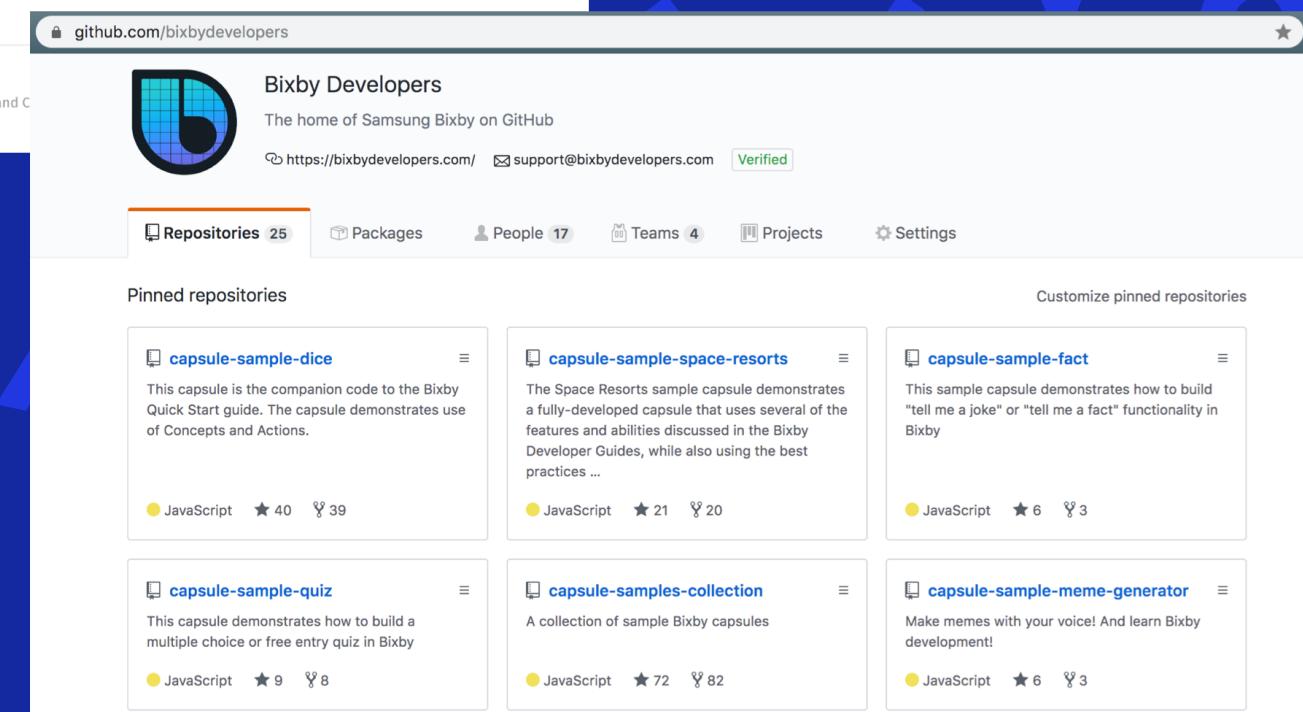
- Core
- Money
- Measurement
- Contact
- Location
- AudioPlayer
- And more!



The Power of Sample Capsules



The screenshot shows the Bixby Developer Center website at bixbydevelopers.com/dev/docs/sample-capsules. The left sidebar is dark-themed and lists various developer resources under 'Samples & Templates' and 'Samples'. The main content area has a light background and features a large title 'Samples & Templates'. Below the title, a paragraph explains that the section contains sample capsules with explanations. A bulleted list defines four types of samples: 'Samples', 'Walkthroughs', 'Templates', and 'Videos'. At the bottom of the page, there's a GitHub link (github.com/bixbydevelopers) and links to privacy policy, terms and conditions, and contact information.



The screenshot shows the GitHub profile for 'Bixby Developers' (github.com/bixbydevelopers). The profile page includes the Bixby logo, a brief description ('The home of Samsung Bixby on GitHub'), and links to the homepage and support email. The navigation bar shows 25 repositories, 17 packages, 4 teams, and 0 projects. The 'Repositories' tab is selected. The page displays six pinned repositories, each with a thumbnail, name, description, and statistics (JavaScript, stars, forks, issues). The repositories listed are: 'capsule-sample-dice', 'capsule-sample-space-resorts', 'capsule-sample-fact', 'capsule-sample-quiz', 'capsule-samples-collection', and 'capsule-sample-meme-generator'.

Now You Have The Power!



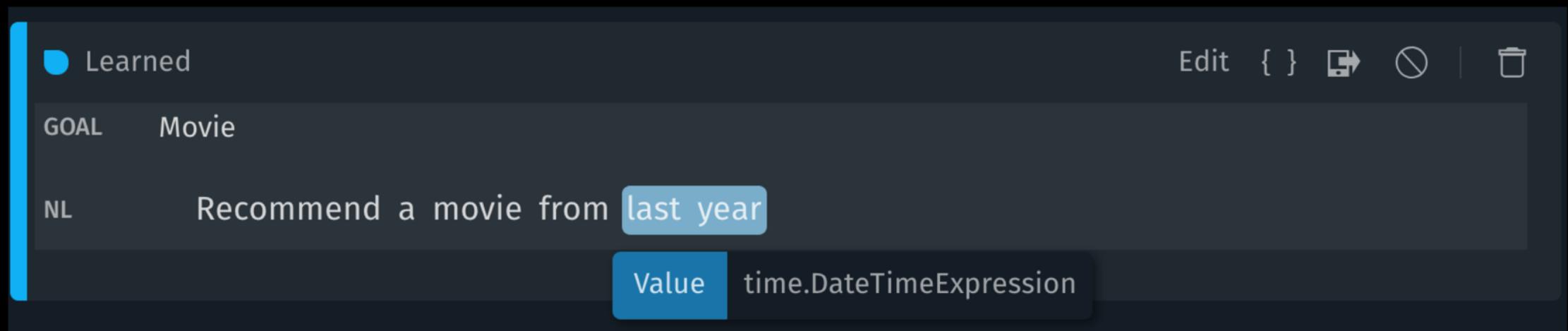


Appendix

How To Use The Time Library?

NL Training

[g:Movie] Recommend a movie from (last year) [v:time.DateTimeExpression]



How To Use The Time Library?

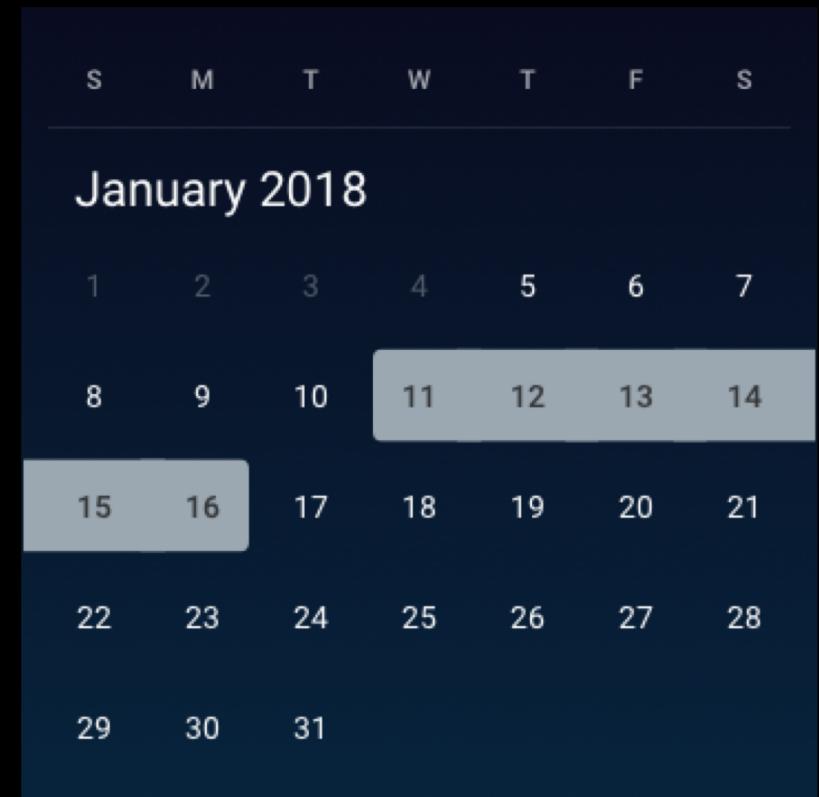
Action Input

```
action (FindMovie) {  
    collect {  
        input (releaseDateTimeExpression) {  
            type (time.DateTimeExpression)  
            min (Optional) max (One)  
            default-select {  
                with-rule {  
                    select-first  
                }  
            }  
        }  
    }  
}
```

How To Use The Time Library?

Views

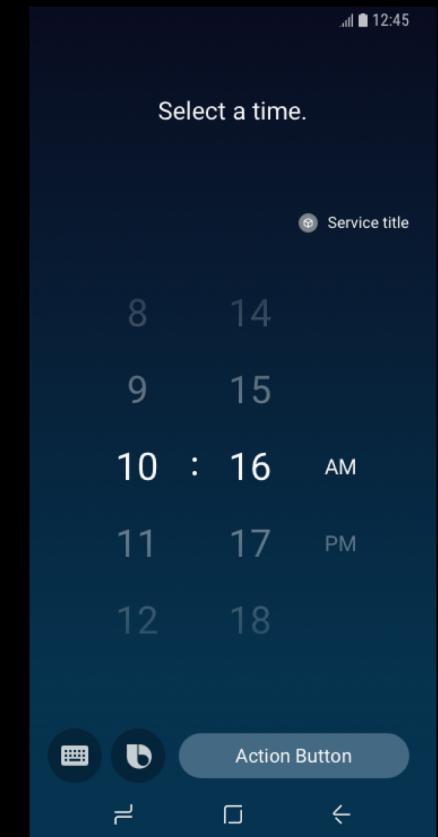
```
input-view {  
  match: DesiredDates (this)  
  render {  
    calendar {  
      allow-range (true)  
      restrictions {  
        blocked-past-dates (false)  
      }  
    }  
  }  
}
```



How To Use The Time Library?

Views

```
input-view {  
    match: DesiredTime (this)  
    render {  
        time-picker {  
            step (10)  
        }  
    }  
}
```



How To Use The Time Library?

Action Implementation

```
// Import JS time module
dates = require('dates')

// From DateTimeExpression
jsDateTime = dates.ZonedDateTime.fromDate(date)
OR
dates.ZonedDateTime.fromDateTime(dateTime)

// From string
jsDateTime = dates.ZonedDateTime.parseDateTime(string, pattern, timezone)

// From user
jsDateTime = dates.ZonedDateTime.now()
```

How To Use The Time Library?

Action Implementation

```
// Calculate  
jsDateTime.durationUntil(other, 'Days')
```

```
// Compare  
jsDateTime.isBefore(other)  
jsDateTime.isAfter(other)
```

```
// Modify  
jsDateTime.plusDays(number)  
jsDateTime.minusYears(number)  
jsDateTime.atStartOfDay()
```

How To Use The Time Library?

Action Implementation

```
// To ISO 8601 string  
stringDateTime = jsDateTime.toISOString()
```

```
// To formatted string  
stringDateTime = jsDateTime.format(pattern)
```

```
// To bxb  
bxbDateTime = jsDateTime.getDateTime()
```

How To Use The Time Library?

Modeling

```
structure (Movie) {  
    property (releaseYear) {  
        type (time.Year)  
        min (Required) max (One)  
        visibility (Private)  
    }  
    property (runtime) {  
        type (time.Period)  
        min (Optional) max (One)  
        visibility (Private)  
    }  
}
```

How To Use The Time Library?

Expression Language

```
// Formatting functions  
value(<time model>)  
dateTime(dateTime, pattern)  
  
// Time functions  
now()  
isFuture(dateTime)  
isPast(dateTime)  
durationBetween(dateTime1, dateTime2)
```