3.2 Контрольные Ссылки

Контрольные Ссылки

Элементы управления в Flet являются объектами, и для доступа к их свойствам нам нужно хранить ссылки (переменные) на эти объекты.

Рассмотрим следующий пример:

import flet as ft

def main(page):

first\_name = ft.TextField(label="Имя", autofocus=True)

last\_name = ft.TextField(label="Фамилия")

greetings = ft.Column()

def btn\_click(e):

greetings.controls.append(ft.Text(f"Привет, {first\_name.value}

{last\_name.value}!"))

first\_name.value = ""

last\_name.value = ""

page.update()

first\_name.focus()

page.add(

first\_name,

last\_name,

ft.ElevatedButton("Поздороваться!", on\_click=btn\_click),

greetings,

)

ft.app(target=main)

В самом начале метода мы создаем три элемента управления, которые

main()

on\_click

TextField

собираемся использовать в обработчике

кнопки: два

для

ввода имени и фамилии и - контейнер для приветственных сообщений. Мы

Column

создаем элементы управления со всеми их свойствами установленными и в конце метода main() , в вызове page.add() , мы используем их ссылки (переменные).

Когда элементов управления и обработчиков событий становится больше и больше, становится сложнее держать все определения элементов управления в одном месте, поэтому они становятся разбросанными по main() . Взглянув на

параметры page.add() , сложно представить, как будет выглядеть конечная форма:

page.add(

first\_name,

last\_name,

ft.ElevatedButton("Поздороваться!", on\_click=btn\_click),

greetings,

)

Является ли Является ли

greetings

first\_name

**Класс Ref**

(имя) (приветствие)

, установлен ли у него autofocus ? или Column ?

Row

Flet предоставляет утилитарный класс Ref , который позволяет определить ссылку на элемент управления, использовать эту ссылку в обработчиках событий и установить ссылку на реальный элемент управления позже, при построении дерева.

TextField

Идея взята из React.

Чтобы определить новую типизированную ссылку на элемент управления:

first\_name = ft.Ref[ft.TextField]()

Для доступа к ссылочному элементу управления (дереференцирование элемента управления) используйте свойство Ref.current :

# незаполненное значение first\_name

first\_name.current.value = ""

Чтобы назначить элемент управления на ссылку, установите

свойство на ссылку таким образом:

Control.ref

page.add(

ft.TextField(ref=first\_name, label="Имя", autofocus=True)

)

*ЗАМЕЧАНИЕ*

*У всех элементов управления Flet есть свойство ref .*

Мы могли бы переписать нашу программу, используя эти ссылки:

import flet as ft

def main(page):

first\_name = ft.Ref[ft.TextField]()

last\_name = ft.Ref[ft.TextField]()

greetings = ft.Ref[ft.Column]()

def btn\_click(e):

greetings.current.controls.append(

ft.Text(f"Привет, {first\_name.current.value}

{last\_name.current.value}!")

)

first\_name.current.value = ""

last\_name.current.value = ""

page.update()

first\_name.current.focus()

page.add(

ft.TextField(ref=first\_name, label="Имя", autofocus=True),

ft.TextField(ref=last\_name, label="Фамилия"),

ft.ElevatedButton("Поздороваться!", on\_click=btn\_click),

ft.Column(ref=greetings),

)

ft.app(target=main)

Теперь в мы ясно видим структуру страницы и все элементы

управления, из которых она состоит.

page.add()

Да, логика становится немного более многословной, так как вам нужно

добавить для доступа к элементу управления ref , но это вопрос

.current.

личного предпочтения :)