

# Велосипед



Дмитрий Игнатъев

# За что мы любим Common Lisp

---

- Макросы
- CLOS
- Интерактивная разработка
- Производительность
- И много за что еще

# За что мы не любим CL?

---

- На нем никто не пишет
- Устаревшая стандартная библиотека
- Мало сторонних библиотек

# За что мы не любим CL?

---

- На нем никто не пишет
- Устаревшая стандартная библиотека
- Мало сторонних библиотек
  - Потому что на нем никто не пишет

# За что мы не любим CL?

---

- На нем никто не пишет
- Устаревшая стандартная библиотека
- Мало сторонних библиотек
  - Потому что на нем никто не пишет
    - Проблема курицы и яйца

# Как нам обустроить Лисп?

---

- Написать свои библиотеки

# Как нам обустроить Лисп?

---

- Написать свои библиотеки
- Написать свой лисп

# Как нам обустроить Лисп?

---

- Написать свои библиотеки
- Написать свой лисп
  - Или хотя бы Clojure



# Как нам обустроить Лисп?

---

- Написать свои библиотеки
- Написать свой лисп
  - Или хотя бы Clojure
  - Теперь у нас 2 лиспа, на которых никто не пишет

# Как нам обустроить Лисп?

---

- Написать свои библиотеки
- Написать свой лисп
  - Или хотя бы Clojure
  - Теперь у нас 2 лиспа, на которых никто не пишет
- Написать велосипед

# Велосипед

---

- <https://github.com/Lovesan/bike>

Common Lisp

.Net Core 2+

# Велосипед

---

□ <https://github.com/Lovesan/bike>

□ **Новый RDNZL**

■ [https://common-lisp.net/~loliveira/ediware/rdnzl/\\_darcs/current/doc/](https://common-lisp.net/~loliveira/ediware/rdnzl/_darcs/current/doc/)

# Велосипед

---

- <https://github.com/Lovesan/bike>
- Новый RDNZL
  - [https://common-lisp.net/~loliveira/ediware/rdnzl/\\_darcs/current/doc/](https://common-lisp.net/~loliveira/ediware/rdnzl/_darcs/current/doc/)
- Интероперабельность vs базовая платформа

# Велосипед

---

- <https://github.com/Lovesan/bike>
- Новый RDNZL
  - [https://common-lisp.net/~loliveira/ediware/rdnzl/\\_darcs/current/doc/](https://common-lisp.net/~loliveira/ediware/rdnzl/_darcs/current/doc/)
- Интероперабельность vs базовая платформа
- Без C++

# Велосипед

---

- <https://github.com/Lovesan/bike>
- Новый RDNZL
  - [https://common-lisp.net/~loliveira/ediware/rdnzl/\\_darcs/current/doc/](https://common-lisp.net/~loliveira/ediware/rdnzl/_darcs/current/doc/)
- Интероперабельность vs базовая платформа
- Без C++
- Кроссплатформенность

□ (ql:quickload :bike)

□ Callbacks

□ Интеграция с conditions

□ Свой DLR

□ И многое другое

□ См. bike-examples

```
(defun bike-examples:start-hello-host
  (&optional (server-port *default-hello-server-port*))
  (declare (type (integer 1 65535) server-port))
  "Starts Kestrel server listener on the specified SERVER-IP"
  ;; Ensure our host is not running first
  (when *host* (bike-examples:stop-hello-host))
  (let ((builder (new 'WebHostBuilder)) ;; create a host builder
        (*server-port* server-port))
    ;; Now we need to do two things, first - configure Kestrel as
    ;; our server.
    ;; Second - configure services available through ASP.NET dependency injections
    (setf builder (invoke 'HostingAbstractionsWebHostBuilderExtensions 'UseContentRoot
                          builder *deps-directory*)
              builder (invoke 'WebHostBuilderKestrelExtensions 'UseKestrel builder
                              (new '(Action KestrelServerOptions)
                                    'configure-kestrel))
              builder (invoke builder 'ConfigureServices
                              (new '(Action WebHostBuilderContext IServiceCollection)
                                    'configure-services)))
    ;; Build the host and start it
    (let ((host (invoke builder 'Build)))
      (invoke host 'Start)
      (setf *host* host))))
```

□ TODO:

- WPF интерфейс
- Обертка над dotnet cli
- Обертка над nuget
- reader-макросы
- Extension-методы
- async/await

□ Help is welcome!



# Спасибо!

---

- Здесь могла бы быть ваша реклама