



Co-marketing to Power your growth with the Akamai Partner Program

```
request := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count}; cc := msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")), count); result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); type ControlMessage struct { Target string; Count int64; }; func main() { controlChannel := make(chan chan bool); workerCompleteChan := make(chan bool); statusPollChannel := reqChan; timeout := time.After(time.Second); select { case result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); controlChannel := make(chan chan bool); workerActive := false; go admin(controlChannel, statusPollChannel); for { select { case respChan := <- statusPollChannel: respChan <- workerActive; case msg := <- controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- request { hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count}; cc := msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")), count); result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } } http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) { reqChan := make(chan chan bool); return; case <- timeout: fmt.Fprintf(w, "TIMEOUT"); }); log.Fatal(http.ListenAndServe(":1337", nil)); } } package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); controlChannel := make(chan chan bool); workerCompleteChan := make(chan bool); statusPollChannel := reqChan; timeout := time.After(time.Second); select { case result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); controlChannel := make(chan chan bool); workerActive := false; go admin(controlChannel, statusPollChannel); for { select { case respChan := <- statusPollChannel: respChan <- workerActive; case msg := <- controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- request { hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count}; cc := msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")), count); result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } } http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) { reqChan := make(chan chan bool); return; case <- timeout: fmt.Fprintf(w, "TIMEOUT"); }); log.Fatal(http.ListenAndServe(":1337", nil)); } } package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); controlChannel := make(chan chan bool); workerCompleteChan := make(chan bool); statusPollChannel := reqChan; timeout := time.After(time.Second); select { case result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); controlChannel := make(chan chan bool); workerActive := false; go admin(controlChannel, statusPollChannel); for { select { case respChan := <- statusPollChannel: respChan <- workerActive; case msg := <- controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- request { hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count}; cc := msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")), count); result := <- reqChan: if result { fmt.Fprintf(w, "ACTIVE"); } else { fmt.Fprintf(w, "INACTIVE"); } time := time.Now(); } } http.HandleFunc("/status", func(w http.ResponseWriter, r *http.Request) { reqChan := make(chan chan bool); return; case <- timeout: fmt.Fprintf(w, "TIMEOUT"); }); log.Fatal(http.ListenAndServe(":1337", nil)); } }
```


Marketing growth with Akamai

Because together, we're stronger

The Akamai Partner Program (APP) isn't just another channel program. It has been designed to empower our partner community with tools and support to unleash your potential and help you grow.



Unlock more profit by reducing cost, increasing productivity, and raising awareness of Akamai products.



Focus on aligning marketing strategies with a best-in-class market development funds (MDF) program.



Deliver co-branded marketing campaigns to drive awareness and sales pipeline, funded by MDF.



*As part of a larger support program, we consider **MDF** to be a key benefit that enables partners to grow with us and deliver innovative and effective marketing plans.*

*Akamai offers **MDF** to qualifying partners (Select, Premier, or Elite) within the APP.*

Akamai Campaign Builder

Tailor-made marketing campaigns

Akamai Campaign Builder (ACB) provides a rich choice of professional marketing collateral and campaign tools to create and launch co-branded marketing campaigns.



Get the right message with minimum effort and maximum impact.



Campaigns are fully subsidized by Akamai and able to be launched globally – at speed.



Increase demand generation and awareness by creating and executing co-branded marketing campaigns.



Readily available, diverse, and customizable resources and content.



Expert ongoing guidance by choosing the right content, planning tactics, and strategies.

Resources and tools

Power your growth

Stay in touch with Akamai through our quarterly partner newsletter, monthly product update webinars, and regular events — including our World Tour and Partner Summits.

Our partner portal provides access to the latest enablement content, webinars, demos, product information, and more. Discover our [partner portal](#).

Make sure you follow us on LinkedIn for the latest news and updates. Check our [LinkedIn showcase page](#) for partners.

Explore and grow your expertise with [Learn Akamai](#).

Trusted support



Our dedicated channel marketing team helps you to generate pipeline and demand. Marketing engagement is proportionate to your partner tier, but you are always supported, regardless of your program level.



We enjoy marketing planning and strategic alignment with Akamai on our marketing investments. It's a valuable function characterized by collaborative input and important campaign feedback from the Akamai marketing team.

– Michael Shrader, Vice President,
Intelligence and Innovative Solutions,
Carahsoft Technology Corp.

Power your growth

Want to find out more about the power of co-marketing with Akamai? Please visit akamai.com/partners.

Akamai – the cloud company that powers and protects life online.

akamai.com