## TestMeasuringJs IntegrationTestCase + setup() + run() + teardown() + page(name:String):HtmlPage + test call api after 5 events() + teardown() + test events get sent() + webclient() + page measure() + webclient(client:WebClient) TestCase + cookie(name:String) + page url(name:String) + page test event recording() + inspect(s:String) + eg(a:Object, b:Object) + eq(a:int, b:int) WebappTest + eq(a:Integer, b:Integer) + eg(a:int, b:int, message:string) + setup() + eq(a:Integer, b:Integer, message:string) + teardown() + eq(a:long, b:long) + test new app gets its metrics() + eq(a:long, b:long, message:string) + eq(a:Long, b:Long) + eq(a:Long, b:Long, message:string) MetricTest + eg(a:boolean, b:boolean) + not eq(a:int, b:int) + setup:void() + not eq(a:long, b:long) + teardown() + eq(a:String, b:String) + test metric installs\_itself() + eq(a:String, b:String, message:string) + test metric installs itself once and only once() + not eq(a:String, b:String) + test processing metrics creates statistic() + assert false(bool:boolean) + test make statistic of one clickrecord() + assert false(bool:boolean, message:string) + test make statistic of more clickrecords() + assert raise(block:Runnable) + test clicks should be processed only once() + assert no raise(block:Runnable) + test make statistics get added to() + assert exception(block:Runnable) + assert no exception(block:Runnable) + pass() + fail(message:string) SiteMenuTest + assert(o:Object) + assert(o:Object, message:string) + test single item() + assert(bool:boolean, message:string) + test multiple items() + assert(bool:boolean) + test subitems() + self.register(test case:TestCase) + initialize() + setup:void() ArrayHelperTest + teardown:void() + run(params:RequestEvent) + test sort of strings() + params(params:RequestEvent) + test sort of ints() + params() + test grouping() + run() + format duration(ms:long) + p (what:long) + p(what:Object) EventDefinitionTest + p(what:String) + test same single node paths match() + pretty exception(ex:Throwable) + test single node subset path matches() + render exception(ex:Throwable) + test\_multiple\_node\_subset\_path matches() + internal line?(line:String) + test not matching number node subset path matches() + test different path node does not match() + test missing path node does not match() + test path matching with html()

+ test real world example1()